



UNIVERSIDAD POLITÉCNICA DE MADRID

ESCUELA TÉCNICA SUPERIOR DE INGENIERÍA Y
DISEÑO INDUSTRIAL

Grado en Ingeniería Electrónica Industrial y Automática

TRABAJO FIN DE GRADO

DISEÑO Y CONSTRUCCIÓN DE UN
HELICÓPTERO COAXIAL CONTROLADO
SOBRE ANDROID

Nicolás Parra Sánchez

Cotutor: D. Alberto Brunete González

Departamento: Electrónica, Automática
e Informática Industrial

Tutor: D. Miguel Hernando Gutiérrez

Departamento: Electrónica, Automática
e Informática Industrial

Madrid, Septiembre 2015



UNIVERSIDAD POLITÉCNICA DE MADRID

ESCUELA TÉCNICA SUPERIOR DE INGENIERÍA Y
DISEÑO INDUSTRIAL

Grado en Ingeniería Electrónica Industrial y Automática

TRABAJO FIN DE GRADO

DISEÑO Y CONSTRUCCIÓN DE UN
HELICÓPTERO COAXIAL CONTROLADO
SOBRE ANDROID

Firma Autor

Firma Cotutor

Firma Tutor

Título: Diseño y construcción de un helicóptero coaxial controlado sobre Android

Autor: Nicolás Parra Sánchez

Tutor: D. Miguel Hernando Gutiérrez

Cotutor: D. Alberto Brunete González

EL TRIBUNAL

Presidente:

Vocal:

Secretario:

Realizado el acto de defensa y lectura del Trabajo Fin de Grado el día 16 de Septiembre de 2015 en Madrid, en la Escuela Técnica Superior de Ingeniería y Diseño Industrial de la Universidad Politécnica de Madrid, acuerda otorgarle la CALIFICACIÓN de:

VOCAL

SECRETARIO

PRESIDENTE

Agradecimientos

Agradezco a mi familia el apoyo incondicional durante la realización de mis estudios y la confianza depositada en mí.

A Eduardo Ortega, compañero de este trabajo, por el gran esfuerzo dedicado y el acompañamiento continuo en la realización del proyecto.

A Miguel Hernando, tutor del proyecto, por servir de guía con su conocimiento y por la gran dedicación a este trabajo.

A Alberto Brunete, cotutor del trabajo, su ayuda prestada.

Resumen

El presente proyecto sienta las bases para el desarrollo de un helicóptero coaxial autónomo. Como principales novedades, se quiere destacar el manejo y control de este. El manejo del helicóptero se consigue desplazando el centro de gravedad. Por otro lado, el control se realiza mediante los sensores de un Smartphone a bordo de la aeronave. Este teléfono además, proporcionará una amplia gama de recursos para el desarrollo de futuras aplicaciones, como pueden ser la cámara o GPS. También se desarrolla la aplicación para enviar órdenes desde el exterior para maniobrar el helicóptero.

Este trabajo se lleva a cabo conjuntamente con mi compañero Eduardo Ortega Biber (1), quién se enfoca en las tareas de diseño y simulación. Mientras que el actual proyecto, se centra en el desarrollo de las dos aplicaciones Android de los teléfonos.

Abstract

This project lays the foundation for the development of an autonomous coaxial helicopter. As main developments, we can highlight the management and control of the helicopter. The handling of the helicopter is achieved by moving the center of gravity. On the other hand, the control is performed by the sensors of a smartphone mounted aboard the aircraft. This phone also provides a wide range of resources for the development of future applications, such as a camera or the GPS sensor. A second application that sends commands from the outside to maneuver the helicopter is also developed.

This thesis is carried out in conjunction with Eduardo Ortega Biber (1), who focuses on the design and simulation tasks. While the current project approaches the development of the two Android phones applications.

Índice general

Agradecimientos	VII
Resumen	IX
Abstract	XI
1. Introducción	1
1.1. Motivación del proyecto	1
1.2. Objetivos	2
1.3. Materiales utilizados.....	2
1.4. Estructura del documento	3
2. Estado del arte	6
2.1. Vehículos aéreos no tripulados	6
2.2. Helicóptero coaxial	7
2.3. Proyectos relacionados	9
3. Descripción Componentes	15
3.1. Motores	15
3.2. Hélices	17
3.3. Controladores electrónicos.....	17
3.4. Batería	18
3.5. Arduino ®	20
3.5.1. ZUM BT-328.....	20
3.6. Servomotores	21
3.7. Sensor ultrasónico	22
3.8. Smartphones.....	23
3.8.1. OnBoard.....	23
3.8.2. Mando externo	24
3.9. Requisitos y especificaciones de componentes.....	24
4. Desarrollo del helicóptero	29
4.1. Sistemas de fabricación	29

4.1.1.	Fresado CNC	29
4.1.2.	Impresión 3D.....	31
4.2.	Banco de pruebas	33
4.3.	Conexión de los componentes.....	35
4.4.	Arduino Yún	37
4.5.	Placa PCB.....	39
4.6.	Placa BQ ZUM BT-328.....	40
4.7.	Configuración, manejo y control.....	42
4.8.	Pruebas y experimentos.....	42
5.	Diseño Final	46
6.	Aplicaciones Android	50
6.1.	Android	50
6.2.	Android Studio.....	51
6.3.	Comunicación	51
6.3.1.	Tecnologías inalámbricas.....	51
6.3.2.	Diagrama global	52
6.3.3.	Protocolo.....	53
6.4.	Test de velocidad	55
6.5.	BTController	57
6.6.	OnBoard Control.....	61
7.	Gestión del proyecto	67
7.1.	Ciclo de vida	67
7.2.	Planificación	68
7.3.	Presupuesto	68
7.3.1.	Personal	68
7.3.2.	Material.....	68
7.3.3.	Resumen de costes.....	69
8.	Conclusiones	71
8.1.	Conclusión.....	71
8.2.	Desarrollos futuros	72
9.	Bibliografía	75
10.	Anexos	78
	Aplicación BTController	78
	Aplicación OnBoard Control	79

Índice de Figuras

Figura 1. Cuadricóptero y Helicóptero coaxial	7
Figura 2. Comparación helicóptero convencional con coaxial	8
Figura 3. Plato oscilante del helicóptero lama	9
Figura 4. Mecanismo de desplazamiento del centro de gravedad.....	10
Figura 5. Helicóptero micro UAV	10
Figura 6. Prototipo helicóptero con desplazamiento del centro de gravedad	11
Figura 7. Montaje de las aspas con bisagras	12
Figura 8. Helicóptero Gimball de Flyability	12
Figura 9. Plataforma Flone para teléfonos móviles.....	13
Figura 10. Motores con rotores coaxiales	16
Figura 11. Motor tipo <i>outrunner</i>	17
Figura 12. Controlador electrónico	18
Figura 13. Batería LiPo.....	19
Figura 14. Placa BQ ZUM BT-328.....	20
Figura 15. Servomotor	21
Figura 16. Sensor de ultrasonidos BAT	22
Figura 17. Smartphone Android	24
Figura 18. Diseño piezas en AutoCAD	30
Figura 19. Impresora 3D BQ Prusa i3 Hephestos	31
Figura 20. Diseño carcasa de la batería	32
Figura 21. Diseño primer banco de pruebas.....	33
Figura 22. Fotografía banco de pruebas	34
Figura 23. Segundo banco de pruebas	35
Figura 24. Conexión de conector tipo servo.....	36
Figura 25. Esquema de conexiones	37
Figura 26. Arduino Yún y librería Bridge	38
Figura 27. Interfaz gráfica en HTML.....	39
Figura 28. Placa PCB.....	40
Figura 29. Interfaz de la aplicación ArduinoRC	41

Figura 30. Desplazamiento del helicóptero variando su centro de gravedad.....	42
Figura 31. Prueba componentes.....	43
Figura 32. Movimiento del sistema mediante el desplazamiento del centro de gravedad.....	43
Figura 33. Ensayo de vuelo en la jaula.....	44
Figura 34. Diseño final helicóptero coaxial	46
Figura 35. Mecanismo para desplazar el centro de gravedad.....	47
Figura 36. Junta cardan	47
Figura 37. Soporte del teléfono móvil.....	48
Figura 38. Fotografía diseño final helicóptero.....	48
Figura 39. Logotipo Android	50
Figura 40. Icono Android Studio	51
Figura 41. Diagrama comunicaciones.....	53
Figura 42. Composición de un mensaje del protocolo	54
Figura 43. Prueba protocolo de comunicación mediante Bluetooth	55
Figura 44. Medida tiempo de envío	56
Figura 45. Gráfica tiempo de envío	56
Figura 46. Diagrama BTController.....	57
Figura 47. Primera actividad BTController.....	58
Figura 48. Actividad de control BTController.....	58
Figura 49. Código lectura acelerómetro	59
Figura 50. Ejes de coordenadas Smartphone	60
Figura 51. Envío de Pitch y Roll BTController.....	60
Figura 52. Diagrama OnBoard Control	61
Figura 53. Interfaz inicio OnBoard Control.....	61
Figura 54. Interfaz actividad de control	62
Figura 55. Diagrama funciones OnBoard Control.....	63
Figura 56. Código para conocer la orientación del Smartphone.....	64
Figura 57. Diagrama de Gantt con la planificación del proyecto.....	68

Índice de Tablas

Tabla 1. Especificaciones técnicas Motores.....	16
Tabla 2. Especificaciones técnicas Controladores electrónicos	18
Tabla 3. Especificaciones técnicas Batería.....	19
Tabla 4. Especificaciones técnicas Microcontrolador	21
Tabla 5. Especificaciones técnicas Servomotor	22
Tabla 6. Especificaciones técnicas Sensor Ultrasónico	23
Tabla 7. Peso de los componentes	25
Tabla 8. Oferta motores coaxiales	25
Tabla 9. Oferta baterías 3S1P	26
Tabla 10. Oferta controladores electrónicos de 25A.....	27
Tabla 11. Especificaciones técnicas Arduino Yún.....	38

Capítulo 1

Introducción

1.1. Motivación del proyecto

El principal motivo para la elección del presente Trabajo Fin de Grado es el interés por realizar un proyecto multidisciplinar, poniendo en práctica todos los conocimientos adquiridos durante el plan de estudios, pero también para estudiar tecnologías y utilizar herramientas que no están en él.

Para profundizar en las asignaturas de electrónica y robótica del Grado, surge un especial interés en la plataforma Arduino, que no tuve la oportunidad de poner en práctica durante la carrera.

Del mismo modo, para continuar mis conocimientos de programación, el desarrollo de aplicaciones en Android me llamó especialmente la atención.

El auge actual de los vehículos aéreos no tripulados provoca también la predilección por esta rama de la robótica. El descenso en el precio de los sensores y la llegada de rápidos microcontroladores a precios reducidos ha permitido que esta especialidad abandone los centros de investigación y pueda ser estudiada sin necesitar grandes inversiones. Además cuenta con la integración de las principales materias estudiadas como la informática, la regulación automática y la electrónica.

Por otro lado, la idea de innovar con el trabajo y buscar un nuevo enfoque a los populares cuadricópteros origina la idea del helicóptero coaxial. Al no estar tan divulgado su desarrollo, supone un desafío aún mayor.

Por último, al ser un producto nuevo, se tiene en mente que sirva como base para el desarrollo de nuevas aplicaciones y funcionalidades al helicóptero. La incorporación de un Smartphone a bordo abre la posibilidad de utilizar los propios sensores del terminal y la exploración de una nueva forma de control.

1.2. Objetivos

Los principales objetivos son, como muestra el título del trabajo, el diseño y construcción de un helicóptero coaxial controlado por el desplazamiento del centro de gravedad mediante una aplicación Android. Para alcanzarlos se tienen que completar los siguientes objetivos:

- Selección de los componentes físicos necesarios para la construcción de un helicóptero coaxial.
- Diseño y construcción de un banco de pruebas que sirva como base para realizar ensayos con los componentes.
- Diseño y construcción de un helicóptero coaxial.
- Desarrollo de las aplicaciones Android que permitan el control de la aeronave. Este punto tiene diferentes objetivos:
 - Control remoto desde el mando externo.
 - Comunicación con protocolo seguro y a alta frecuencia entre Arduino y el móvil a bordo.
 - Control de actitud en el Smartphone OnBoard.

1.3. Materiales utilizados

Durante la elaboración de este trabajo se han empleado diferentes instrumentos proporcionados por el Departamento de Electrónica, Automática e Informática Industrial. Entre ellos se encuentran las herramientas utilizadas para la fabricación del helicóptero, como la impresora Hephestos i3 y la máquina fresadora CNC. La impresora se emplea para producir las piezas en plástico PLA, mientras que las piezas de metacrilato se realizan con la fresadora. Para construir el helicóptero se aprovechan las herramientas, llaves y demás utensilios del mencionado departamento.

Se utilizan diferentes herramientas de software ejecutadas en un ordenador personal, con el sistema operativo Windows 7. Los principales programas utilizados son AutoCAD y Autodesk Inventor, para las tareas de creación de diseños, y Arduino IDE y Android Studio, para el desarrollo de las aplicaciones Arduino y Android respectivamente.

Por supuesto, para el desarrollo y ejecución de las aplicaciones Android se utilizan dos Smartphones personales.

1.4. Estructura del documento

A continuación, y para facilitar la lectura del documento, se detalla el contenido de cada capítulo.

- En el primer capítulo se introduce al lector en el proyecto, indicando los motivos de su elección y los objetivos a desarrollar.
- En el segundo capítulo se explica la evolución de los vehículos aéreos no tripulados, describiéndose las características de su variante coaxial. Así mismo, se exponen proyectos relacionados con el presente trabajo.
- En el tercer capítulo se describen los componentes utilizados para la construcción del helicóptero.
- En el cuarto capítulo se relata el desarrollo del helicóptero, detallando todos los procesos hasta su consecución final. En el último punto se describen las pruebas y experimentos llevados a cabo.
- En el quinto capítulo se muestra el diseño final del helicóptero coaxial, detallando las diferentes partes que lo componen.
- En el sexto capítulo se describen las aplicaciones Android desarrolladas para los dos Smartphones, el que está a bordo y el mando exterior.

- En el séptimo capítulo se detalla el ciclo de vida del proyecto y se desglosa su presupuesto.
- El octavo capítulo sirve para mostrar las conclusiones y presentar posibles desarrollos futuros del proyecto.

Capítulo 2

Estado del arte

En este capítulo se introducen los vehículos aéreos no tripulados y se describe el concepto de helicóptero coaxial junto con sus ventajas, comparándolo a diferentes configuraciones de helicópteros. Una vez establecido el marco, el estado del arte se refleja mostrando los proyectos relacionados con el trabajo actual.

2.1. Vehículos aéreos no tripulados

Los vehículos aéreos no tripulados, o UAV por sus siglas en inglés, (Unmanned Aerial Vehicle) son aeronaves que vuelan sin tripulación humana a bordo. El avión no tripulado más antiguo fue desarrollado después de la Primera Guerra Mundial y se empleó durante la Segunda Guerra Mundial para entrenar a los operarios de los cañones antiaéreos (2).

Históricamente, los vehículos aéreos no tripulados eran principalmente aviones pilotados de forma remota y usados con fines militares, tanto de reconocimiento como de ataque, donde son denominados vehículos aéreos de combate no tripulados (UCAV).

Con el avance de la tecnología de los microprocesadores y la mejora de las capacidades de las baterías ha aumentado el desarrollo de este tipo de vehículos. En la actualidad se utilizan, cada vez más, en aplicaciones civiles de vigilancia, seguridad e incluso educativas y recreativas.

Los vehículos aéreos se pueden clasificar en dos grandes divisiones: de ala fija o de ala giratoria. Dentro de la categoría de ala giratoria se puede definir un helicóptero como una aeronave de alas rotatorias, más pesada que el aire, cuyo

rotor gira por la acción de una fuente de potencia produciendo su desplazamiento y su sustentación. Estos, a su vez, se pueden diferenciar por el número de rotores y su colocación (tándem, coaxiales, lado a lado, etc...) (3).

La gran ventaja de los helicópteros reside en su capacidad para generar sustentación sin que la aeronave se desplace, permitiendo, por tanto, despegar y aterrizar de manera vertical (VTOL - Vertical Take-Off and Landing).

En los últimos años, la mayor parte del desarrollo de helicópteros no tripulados se ha centrado en su variante de cuatro rotores dispuestos en las extremidades de una cruz. La gran popularidad de este tipo de vehículo, conocido como cuadricóptero, se debe a su gran maniobrabilidad y relativa facilidad de control.



FIGURA 1. CUADRICÓPTERO Y HELICÓPTERO COAXIAL

A diferencia del cuadricóptero, los helicópteros coaxiales son aeronaves con dos rotores que giran en sentido opuesto sobre un mismo eje. Esta característica le permite consumir menos potencia para una carga dada y ser construido de manera más compacta en comparación con un cuadricóptero. No obstante, tiene la desventaja de ser mecánicamente más complejo, debido que el eje principal que tiene que proporcionar propulsión y control, de forma simultánea, a los dos rotores que giran en sentido opuesto.

2.2. Helicóptero coaxial

La particularidad del helicóptero coaxial se fundamenta en el uso de dos rotores en contra rotación, que compensan y anulan el par resultante sobre el

fuselaje de la aeronave. Esto le permite poder prescindir del rotor de cola o anti-par, necesario en los helicópteros convencionales para contrarrestar el par generado por el rotor principal. Al tener dos rotores principales, toda la potencia se emplea en generar sustentación, aumentando así la eficiencia energética de estos vehículos. En ensayos experimentales se ha demostrado que con un diseño coaxial se consume un 5% menos de potencia en vuelo estático que con el mismo empuje en un helicóptero con un solo rotor principal (4).

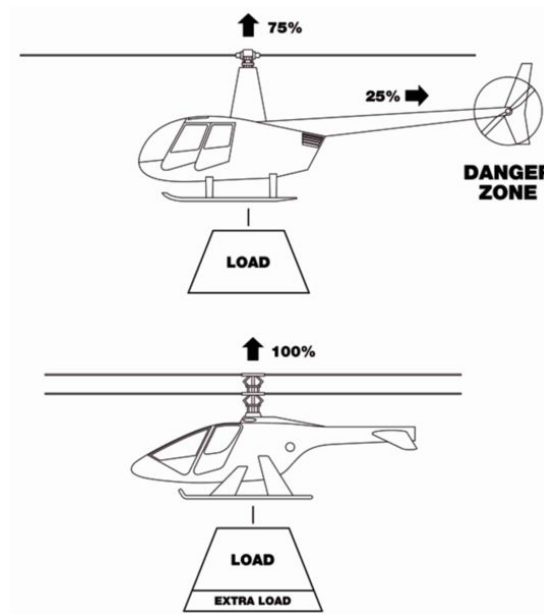


FIGURA 2. COMPARACIÓN HELICÓPTERO CONVENCIONAL CON COAXIAL

Al no necesitar un rotor de cola se consigue una estructura más compacta pudiendo reducir el tamaño del helicóptero un 35 - 40% en comparación con un helicóptero convencional. Esto implica a su vez una reducción del momento de inercia del vehículo, mejorando así su control y manejabilidad (5).

Otra desventaja de los helicópteros convencionales, que se evita con un diseño coaxial, es la asimetría de la velocidad de giro de las aspas. En vuelo de avance, el aspa de la hélice que se mueve en el sentido de avance del helicóptero va a mayor velocidad que el aspa que retrocede, provocando así dicha asimetría de la sustentación. En un helicóptero coaxial este efecto se anula debido a que los dos rotores principales giran en sentido opuesto (6).

El principal inconveniente es la necesidad de incorporar un mecanismo para controlar la aeronave. Al contrario que en los cuadricópteros, en los que el alabeo y el cabeceo se obtienen aumentando y disminuyendo la velocidad de sus rotores,

un helicóptero requiere un dispositivo auxiliar para realizar dichas maniobras. Como consecuencia, suele ser imprescindible añadir dos actuadores adicionales.

2.3. Proyectos relacionados

En este punto se presenta la literatura relacionada con el análisis de estas aeronaves.

En el artículo (7) se realiza un modelo dinámico del helicóptero coaxial Lama y se compara con un modelo de un cuadricóptero para apreciar las diferencias entre la dinámica y el control de ambos vehículos. El helicóptero Lama se distribuye como aeromodelismo de ocio y es controlado por radio. Para maniobrar utiliza un plato oscilante, parecido al de los helicópteros convencionales, accionado por servomotores. Además dispone de un giroscopio para proporcionar realimentación en lazo de control de la guiñada.



FIGURA 3. PLATO OSCILANTE DEL HELICÓPTERO LAMA

Un proyecto muy similar al presente trabajo se expone en el artículo (8). Allí se presentan el diseño y el control de un helicóptero coaxial con un mecanismo de desplazamiento del centro de gravedad. El diseño preliminar es capaz de estabilizarse de forma pasiva en el alabeo y cabeceo frente a pequeñas perturbaciones. Cuenta también con un control activo de la altura, así como de la guiñada. Para conseguir el movimiento horizontal de la aeronave se mueve el

peso de la batería mediante el mecanismo de desplazamiento del centro de gravedad mostrado en la siguiente ilustración.

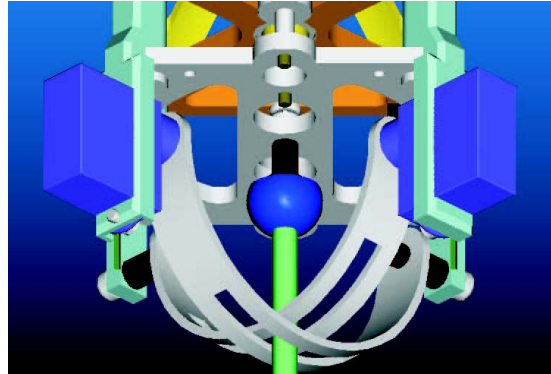


FIGURA 4. MECANISMO DE DESPLAZAMIENTO DEL CENTRO DE GRAVEDAD

Como se puede observar, se utilizan dos servomotores y una articulación esférica. Al final del mencionado artículo se indica que la aeronave es capaz de mantener vuelo estático de sustentación durante periodos cortos de tiempo, en cambio, no es capaz aún de maniobrar en el aire.

La evolución del artículo anterior, se da en el trabajo (9), donde se describe la construcción de otro helicóptero coaxial detallando el control realizado en el alabeo y cabeceo. El helicóptero diseñado pertenece a la categoría de micro UAVs debido a su pequeño tamaño. Se incorporan, en él, sensores infrarrojos para la estimación de altitud, giroscopios e inclinómetros para calcular la posición de la aeronave. Los reguladores utilizados para los controles son del tipo proporcional derivativo (PD). Del mismo modo, se incorpora un mecanismo para el desplazamiento del centro de gravedad muy similar al anterior.

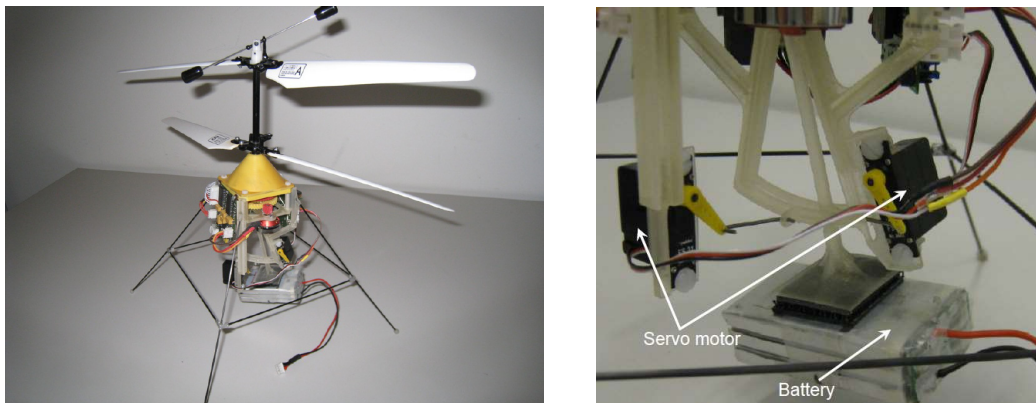


FIGURA 5. HELICÓPTERO MICRO UAV

El desarrollo de un modelo matemático de la dinámica de estos helicópteros se explica en (10). Una vez creado el modelo dinámico se realizan simulaciones que demuestran la viabilidad de las aeronaves frente a diferentes condiciones. También se implementa un control de altura. La tesis concluye con la presentación de diversos prototipos, pero indica que ninguno es capaz de realizar un vuelo autónomo, debido a la falta de fiabilidad del mecanismo de desplazamiento del centro de gravedad y el deterioro prematuro de este después de los primeros ensayos.

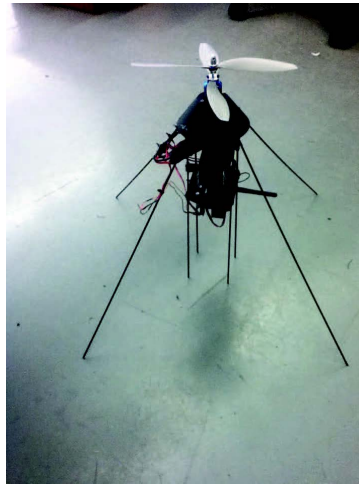


FIGURA 6. PROTOTIPO HELICÓPTERO CON DESPLAZAMIENTO DEL CENTRO DE GRAVEDAD

Un mecanismo, diferente a los mencionados anteriormente, es el desarrollado en (11). En este artículo se diseña un micro helicóptero coaxial que prescinde de cualquier actuador adicional para efectuar el control de actitud de la aeronave.

El control del movimiento del helicóptero se consigue variando de forma cíclica el ángulo de ataque de las aspas. Al contrario que en los helicópteros convencionales, en los que se utilizan dos servomotores y un plato oscilante, en este trabajo se diseña un mecanismo que varía el ángulo de forma pasiva. Las aspas se sujetan a los motores mediante bisagras, permitiendo de esta forma a los álabes cambiar su ángulo de incidencia dependiendo del par generado por el motor.

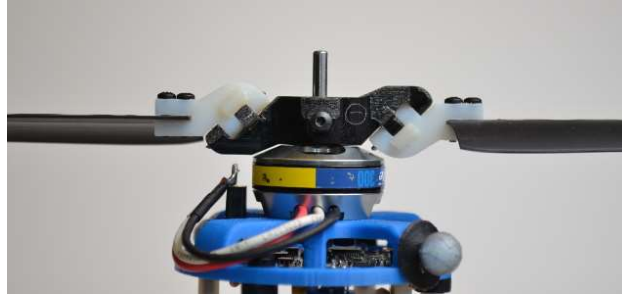


FIGURA 7. MONTAJE DE LAS ASPAS CON BISAGRAS

Así, el diseño mecánico se simplifica y aumenta, como consecuencia, la dificultad de la computación de bajo nivel para gobernar los motores. Al aumentar el par generado por el motor en una sección de su rotación se incrementa el empuje creado por una de sus palas a la vez que se disminuye el empuje de la otra pala. Este aspecto viene determinado por la orientación de las bisagras que permiten a una de ellas incrementar, exclusivamente, su ángulo de incidencia y a la segunda, únicamente, reducir este ángulo.

El proyecto de investigación Gimball (12), ejecutado durante cinco años en la Escuela Politécnica Federal de Lausana en Suiza, ha desarrollado un helicóptero coaxial controlado mediante la desviación del flujo de aire de sus rotores.



FIGURA 8. HELICÓPTERO GIMBALL DE FLYABILITY

Utiliza para ello dos alerones dispuestos en forma de cruz debajo de las hélices. La particularidad de este diseño se encuentra en su protección. Una estructura esférica envuelve la aeronave y le permite sufrir colisiones sin perder la estabilidad. Este vehículo de Flyability es capaz de acceder a terrenos accidentados o construcciones complejas pudiendo desempeñar labores de seguridad y rescate.

Para permitir al operario realizar las maniobras, dispone de una cámara para visualizar la perspectiva del helicóptero.

Dado el objetivo, del presente proyecto, de utilizar un teléfono móvil como sistema sensorial, se encuentra el siguiente proyecto. Flone (13) es una plataforma que monta un teléfono móvil en un cuadricóptero. Este proyecto propone la incorporación del teléfono con la expectativa de que se desarrollen diferentes aplicaciones que aumenten su funcionalidad. Actualmente se encuentra en proceso de desarrollo. En las primeras etapas dispone de un controlador de vuelo Multiwii, basado en Arduino, y una aplicación de grabación de vídeo para el teléfono móvil. Se pretende, en etapas posteriores, prescindir del microcontrolador y así simplificar el diseño.



FIGURA 9. PLATAFORMA FLONE PARA TELÉFONOS MÓVILES

Capítulo 3

Descripción Componentes

En este apartado se describen los componentes utilizados a lo largo del proyecto para el montaje y la utilización del helicóptero.

El primer paso en la fase de diseño, fue el análisis de los requisitos y especificaciones para una correcta elección de los componentes necesarios. A continuación, se enumeran las características de cada elemento y en el último punto se justifica la elección de cada uno teniendo en cuenta el conjunto global.

3.1. Motores

El principal elemento en un helicóptero con un diseño coaxial es, sin lugar a dudas, los motores. Estos son una pareja de rotores girando en direcciones opuestas y ensamblados en el mismo eje de rotación.

En el trabajo se ha utilizado la tipología de motores eléctricos sin escobillas, que prescinden de estas para el cambio de polaridad en el rotor. Esta ausencia provoca un menor rozamiento y, por lo tanto, un mayor rendimiento. A estas ventajas, también se suman un menor mantenimiento y un menor desprendimiento de calor.

Este tipo de motores necesitan una señal trifásica para crear la rotación del campo magnético. Como en nuestro caso, la alimentación se realiza mediante corriente continua, es imprescindible la utilización de un controlador electrónico que genere la señal trifásica.



FIGURA 10. MOTORES CON ROTORES COAXIALES

A continuación se describen las especificaciones técnicas de los motores utilizados, siendo factores clave para la elección final de los componentes la velocidad angular (kV) y la cantidad máxima de empuje.

TABLA 1. ESPECIFICACIONES TÉCNICAS MOTORES

Model:	AEO CR28M
Dimension:	28 x 51 mm
Weight:	79 g
Diameter of shaft:	3.0 mm
Length of front shaft:	F10.25 mm
Max. Current:	$17A \times 2 = 34A$
Max. Power:	375 W
kV:	1050 rpm/V
Propellers:	10x4 or 10x5
Thrust:	1320 g (11 V 10.5A x 2)
ESC Required:	Two 25A

También mencionar, que este tipo de motores son de tipo *outrunner*, en los que la parte móvil es la carcasa exterior, compuesta de imanes, y la parte fija son los bobinados que componen la carcasa interior.



FIGURA 11. MOTOR TIPO *OUTRUNNER*

3.2. Hélices

Las hélices tienen principalmente tres características:

- El número de palas: lo más habitual son las hélices con dos palas, aunque también se pueden encontrar variantes con tres o más álabes.
- El diámetro del círculo que trazan las puntas de las hélices al rotar.
- El paso: es el avance de las palas en cada giro suponiendo que se encontraran en un medio sólido.

Estos dos últimos vectores definen el tipo de hélice y vienen expresados en pulgadas.

Las hélices utilizadas son APC 10 x 4.7 SF.

3.3. Controladores electrónicos

El controlador electrónico, cuyas siglas en inglés son ESC (Electronic Speed Controller), es un circuito electrónico que tiene la función de variar la velocidad de un motor eléctrico y actuar de freno dinámico.

Estos circuitos, como se ha citado anteriormente, son imprescindibles para la utilización de un motor sin escobillas a través de corriente continua.



FIGURA 12. CONTROLADOR ELECTRÓNICO

A través de las señales moduladas por ancho de pulso enviadas desde el microcontrolador, el variador proporciona una señal eléctrica trifásica que generará la rotación del motor.

A continuación se describen las especificaciones de los controladores electrónicos utilizados.

TABLA 2. ESPECIFICACIONES TÉCNICAS CONTROLADORES ELECTRÓNICOS

Model:	Turnigy Plush – 25A
Dimension:	24 x 45 x 11 mm
Weight:	22g
Cont Current:	25A
Burst Current:	35A
BEC:	Linear 5V / 2A
Lipo Cells:	2-4
NiMH :	5-12

3.4. Batería

Las baterías son dispositivos que convierten la energía química almacenada en celdas electroquímicas en energía eléctrica. La batería utilizada es de polímero de litio (LiPo), que es la más utilizada en aeromodelismo por su alta relación potencia-peso y su disminuida variación de tensión. La posibilidad de utilizar una

alta tasa de descarga, la hace ideal para este sector ya que permite alcanzar picos de potencia muy altos en comparación con otro tipo de baterías.

En las baterías de polímero de litio se utiliza la nomenclatura xSyP, donde “x” representa el número de celdas en serie e “y” indica las que se encuentran en paralelo.



FIGURA 13. BATERÍA LIPO

A continuación se describen las especificaciones de la batería utilizada, donde la tasa de descarga es un factor clave en la elección de la misma, como veremos en el último punto del capítulo.

TABLA 3. ESPECIFICACIONES TÉCNICAS BATERÍA

Model:	Rhino 4900 mAh
Capacity:	4900 mAh
Constant discharge:	20C
Burst rate:	30C (15sec)
Configuration:	3S 11.1V
Pack size:	165 x 25 x 41 mm
Weight:	352 g

3.5. Arduino ®

Arduino es una plataforma de hardware libre, basada en una placa con un microcontrolador y un entorno de desarrollo, diseñada para facilitar el uso de la electrónica en prototipos de proyectos multidisciplinarios. Su hardware consiste principalmente en un microprocesador Atmel AVR y puertos de entrada y salida, tanto digitales como analógicas, para la conexión de sensores y actuadores (14).

El microcontrolador en este proyecto es el encargado de recibir los datos por parte del teléfono móvil OnBoard para controlar el helicóptero. Por lo tanto, el requisito mínimo es contar con cuatro canales de salida con modulación por ancho de pulso, para controlar los dos servomotores y los dos variadores de los motores, y tecnología para poder comunicarse con el teléfono móvil. La comunicación es por vía Bluetooth, dada su portabilidad inalámbrica y su alta velocidad.

3.5.1. ZUM BT-328

La placa elegida para desarrollar el trabajo es la ZUM BT-328, de la empresa BQ y totalmente compatible con Arduino. Como principales características, relevantes para este proyecto, podemos diferenciar los conectores con tres pines, para el conexonado directo de los servomotores y controladores electrónicos, un módulo Bluetooth incorporado para la comunicación con el Smartphone y un amplio rango de tensión de entrada, lo que permite la alimentación directamente desde la batería LiPo.

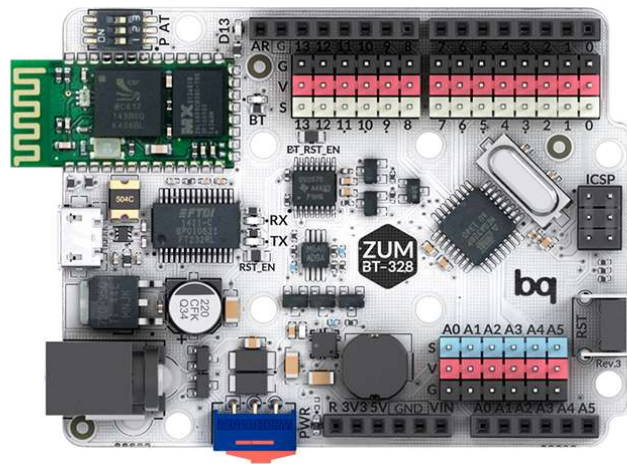


FIGURA 14. PLACA BQ ZUM BT-328

TABLA 4. ESPECIFICACIONES TÉCNICAS MICROCONTROLADOR

Modelo:	BQ ZUM BT-328
Voltaje de entrada:	6 - 17 V
Corriente de salida:	3.3 (50mA) - 5 (3.2A) V
Velocidad de CPU:	16 MIPS
Digital:	14 I/O
Analógico:	6
Interfaces:	Bluetooth TM 2.1 Clase 2
Dimensiones:	67 x 54 mm

3.6. Servomotores

El servomotor es un dispositivo actuador capaz de rotar dentro de un rango de operación. Está formado por un motor de corriente continua con un mecanismo reductor y un circuito de control con un potenciómetro para poder controlar su posición. Para controlar el servomotor es necesario el envío de una señal modulada en ancho de pulso con una frecuencia de 50 Hz. La anchura del pulso, que normalmente se comprende entre uno y dos milisegundos, codifica el ángulo de giro para posicionarse en el ángulo requerido.

En este trabajo, los actuadores tendrán la misión de desplazar el centro de gravedad del helicóptero, para poder controlar el movimiento de este.



FIGURA 15. SERVOMOTOR

A continuación se describen las especificaciones técnicas del servomotor utilizado, siendo el par la característica más relevante.

TABLA 5. ESPECIFICACIONES TÉCNICAS SERVOMOTOR

Torque Rate:	10 Kg/cm
Speed:	.16sec/60deg
Input:	5~7v
Type:	Digital
Gearing:	Metal
Weight:	55g
Size:	40.7 x 19.7 x 42.9mm

3.7. Sensor ultrasónico

Los sensores de ultrasonidos son detectores de proximidad que trabajan libres de roces mecánicos y que detectan objetos a distancias que van desde pocos centímetros hasta varios metros. Este detector emite ondas de ultrasonidos y mide el tiempo que tarda en regresar el eco después de rebotar sobre un objeto.

Para poder controlar la velocidad de ascenso y descenso del helicóptero, es necesario poder medir su altura sobre el suelo, utilizando este sensor para tal fin.

El sensor escogido es el BAT Sonar Range Finder, muy utilizado en aplicaciones de robótica, por su precisión, pequeño tamaño y precio reducido.

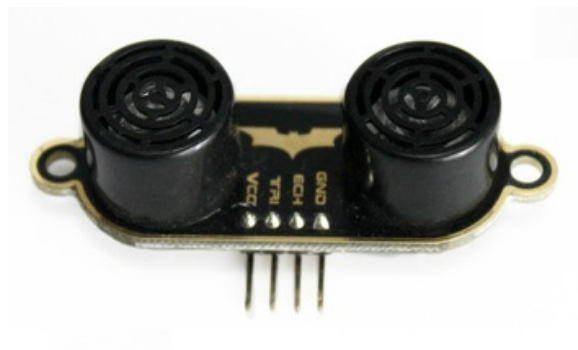


FIGURA 16. SENSOR DE ULTRASONIDOS BAT

TABLA 6. ESPECIFICACIONES TÉCNICAS SENSOR ULTRASÓNICO

Rango medición:	2 – 500 cm
Tensión alimentación:	5 V
Ángulo efectivo:	15°
Resolución:	0,2 cm

3.8. Smartphones

Los Smartphones, o teléfonos inteligentes, son dispositivos que integran las funcionalidades de un teléfono móvil con las características similares a un ordenador personal. Estos permiten una gran conectividad inalámbrica y poseen diferentes sensores, que son de gran utilidad en el montaje de cualquier avión no tripulado, brindando multitud de características e infinitas soluciones a este.

Para la realización del proyecto, se han utilizado dos teléfonos inteligentes para el control y funcionamiento del helicóptero. A continuación se describe cada uno de ellos.

3.8.1. OnBoard

El término OnBoard se refiere, al Smartphone que va a bordo de la aeronave. Este dispositivo recibe las señales del mando externo, y a través de la lectura de sus sensores, envía las órdenes al microcontrolador para variar la posición de los servomotores y la velocidad de los motores.

Este teléfono móvil es el encargado de realizar el control del helicóptero mediante una aplicación Android, ya que a través del acelerómetro y magnetómetro que posee, es capaz de obtener la orientación del terminal, y en consecuencia la de la aeronave, y ejecutar el controlador para conseguir la orientación deseada.

Los requisitos mínimos del dispositivo para hacer las funciones requeridas son disponer de los sensores de acelerómetro y magnetómetro, ser compatible con las tecnologías de comunicación inalámbricas, una velocidad de procesador suficientemente considerable, alrededor de 1 GHz, y disponer del sistema operativo Android lo más actualizado posible para acceder a todas las funcionalidades.



FIGURA 17. SMARTPHONE ANDROID

3.8.2. Mando externo

El Smartphone exterior actúa como mando inalámbrico para enviar y recibir señales al teléfono OnBoard. Los diferentes medios de comunicación y portabilidad de un Smartphone, hace que sea interesante utilizarlo para el envío de señales desde el exterior.

Desde este teléfono, con la aplicación BTController, descrita posteriormente en el capítulo 6, se mandan las consignas de altura, pitch, roll y yaw.

3.9. Requisitos y especificaciones de componentes

En este punto se pretende justificar la elección de cada componente descrito anteriormente, teniendo en cuenta los requisitos y especificaciones de la solución global.

La decisión de uso de cada uno de los elementos, viene supeditada en gran medida por el diseño final y la compatibilidad entre ellos, cumpliendo los requerimientos finales.

En primer lugar, la elección del motor depende directamente de la masa que se necesite levantar. Es necesario saber el peso, o en su defecto una estimación, de los componentes que conforman el helicóptero.

A continuación se muestra la tabla con los pesos de cada elemento, que permite saber el empuje necesario que debe realizar el motor.

TABLA 7. PESO DE LOS COMPONENTES

Componente	Peso
Motores y hélices	79 g
Batería	352 g
Controladores electrónicos	2 x 22 g
Servomotores	2 x 55 g
Microcontrolador	23 g
Smartphone OnBoard	110 g
Peso Total	718 g

Aparte de evaluar el posible peso de la estructura del helicóptero, es necesario establecer un margen para el empuje de los motores que permita al helicóptero maniobrar con agilidad.

Si se observa la oferta de motores coaxiales capaces de levantar el peso calculado, mostrados en la tabla de abajo, se encuentran cuatro opciones. Cada una de ellas tienen respectivamente empujes de 744, 800, 960 y 1320 gramos. Las dos primeras alternativas no ofrecen suficiente potencia y la tercera limita un excedente de peso a 242 gramos. Esta diferencia restringe demasiado el diseño de la estructura, escogiéndose finalmente el último motor.

TABLA 8. OFERTA MOTORES COAXIALES

Modelo	Current (A)	Max Current (A)	Power (W)	Peso (g)	Thrust máximo (g)	ESC	Precio
CR23L Contra Rotating BL System1100kv	17,7	20,02	254	109	960	2 x 25 A	\$ 41,99
CR23M Contra Rotating BL System1550kv	18	21,04	269	73	800	2x 25 A	\$ 42,99
CR23S Contra Rotating BL System 1700kv	19	21	264	64	744	2x 25 A	\$ 43,55
Contra Rotating BL System 375W	21	34	375	79	1320	2 x 25 A	\$ 59,99

Una vez elegido el motor, el siguiente elemento es la batería. Para una duración de vuelo superior a 10 minutos se necesita una capacidad superior a 3500 mAh. Este valor viene determinado por la intensidad nominal de los motores de 21A.

$$Capacidad [mAh] = \frac{10 \text{ min}}{60 \text{ min/h}} \times 21 \text{ A} \times \frac{1000 \text{ mA}}{1 \text{ A}} = 3500 \text{ mAh}$$

Además, para cumplir con las necesidades del motor necesita ser del tipo 3S1P (11,1 V) y tener una constante de descarga suficiente para poder entregar intensidades de 34A. A continuación se muestra la tabla con la oferta de baterías 3S1P.

TABLA 9. OFERTA BATERÍAS 3S1P

Modelo	Capacidad (mAh)	Discharge (C)	I max. (C · mAh) [A]	t max. (mAh/21A) [min]	Peso (g)	Precio
R2350-20-3	2350	20	47	6,71	205	\$ 14,00
R2550-25-3	2550	25	63,75	7,29	190	\$ 12,55
R2550-40-3	2550	40	102	7,29	231	\$ 18,00
R3700-20-3	3700	20	74	10,57	330	\$ 16,49
R3700-25-3	3700	25	92,5	10,57	327	\$ 19,95
R4000-25-3	4000	25	100	11,43	342	\$ 23,40
R4900-20-3	4900	20	98	14,00	352	\$ 22,49
R4900-25-3	4900	25	122,5	14,00	395	\$ 37,13

La batería Rhino 4900mAh 3S1P 20C supera las necesidades descritas y, aunque ligeramente más pesada, es finalmente la elegida.

Las especificaciones del motor requieren dos controladores de 25A; se eligen dos Turnigy Plush 25amp Speed Controller que toleran las intensidades máximas solicitadas.

TABLA 10. OFERTA CONTROLADORES ELECTRÓNICOS DE 25A

Modelo	Peso (g)	Tamaño (mm)	Current (A)	Burst (A)	BEC (A)	Precio
HK-SS30A- HW	24	45 x 25 x 4	25	30	3	\$ 9,98
TURNIGY Plush 25amp	22	24 x 45 x 11	25	35	2	\$ 13,45

No menos importante es la elección de los servomotores, que deben ser capaces de mover el centro de gravedad en ambas direcciones. Con un par de 10 Kg/cm, tabla 5, es suficiente para dicho fin.

Capítulo 4

Desarrollo del helicóptero

En esta fase de desarrollo se detallan algunos de los procesos llevados a cabo para la consecución final del helicóptero.

Con los componentes seleccionados, es conveniente probar su funcionamiento en conjunto para poder identificar posibles defectos o interferencias entre ellos. En consecuencia, esta fase sirve de pruebas y para la familiarización con los componentes y herramientas. Se fabrica un banco de pruebas que sirva como validación terrestre del diseño aéreo final.

4.1. Sistemas de fabricación

En este punto se describen los sistemas utilizados para la fabricación del helicóptero y el banco de pruebas.

4.1.1. Fresado CNC

Para la elaboración de la mayoría de las piezas de la estructura del banco de pruebas, definido en puntos posteriores, se utiliza el fresado CNC, diseñándose en primer lugar un modelo 2D con AutoCAD 2015.

Este software de Diseño Asistido por Computadora o CAD, por sus siglas en inglés, es desarrollado y comercializado por la empresa Autodesk. Una vez diseñadas las piezas que forman la estructura se exportan en formato DXF (acrónimo del inglés Drawing Exchange Format) para que puedan ser interpretadas por el software necesario para su posterior fresado.

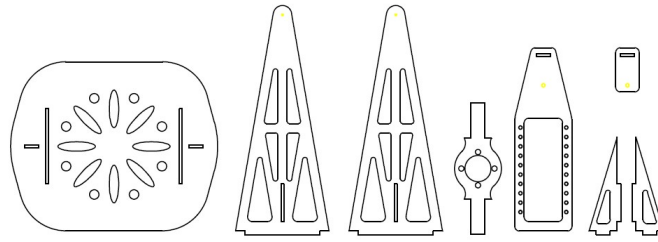


FIGURA 18. DISEÑO PIEZAS EN AUTOCAD

Este software se compone en este caso de dos programas. En primer lugar se utiliza VCarve para calcular las trayectorias de fresado y taladro para el mecanizado en la máquina de control numérico. Una característica muy importante de este programa es la posibilidad de visualizar en la pantalla del ordenador el resultado del diseño. Esta propiedad permite reconocer con rapidez problemas de diseño o configuración de las trayectorias. La forma de trabajar con VCarve es muy intuitiva. En primer lugar, se seleccionan las características del material de trabajo definiéndose el área disponible y su grosor. A continuación se coloca sobre esta superficie el dibujo de la pieza exportado por AutoCAD. Por último, se van seleccionando contornos cerrados de dicho dibujo y tras la elección del método de mecanizado y de la herramienta a utilizar el programa calcula la trayectoria necesaria. En este momento conviene visualizar la trayectoria creada para confirmar que cumple con todos los requisitos y una vez aprobado se exportan las trayectorias a un archivo *.txt*.

La segunda herramienta, Match 5, se utiliza para controlar la máquina fresadora por Control Numérico Computarizado (CNC). Este programa utiliza las trayectorias de herramienta definidas anteriormente para gobernar la fresadora. Cuenta principalmente con una botonera virtual de puesta en marcha, parada, pulsador de emergencia y un teclado para maniobrar manualmente la fresadora. Para empezar a fresar es importante en primer lugar sujetar firmemente la plancha de polimetilmetacrilato sobre la mesa de la máquina de mecanizado. Un método, muy cómodo y efectivo para tal fin, es utilizar cinta de doble cara, asegurándose que la zona de fresado quede inmovilizada. Esto evita que durante el fresado puedan desprenderse trozos de metacrilato, con el consiguiente peligro. A continuación se coloca la fresa escogida en el mandril. En este caso se trata de una fresa universal de 3mm de carburo de tungsteno (carbide). Con la plancha fija y la fresa en su lugar, el siguiente paso es indicar al programa Match 5 las

coordenadas que tomará de origen. Para ello se mueve el cabezal a la zona de metacrilato escogida para el fresado de la pieza. Es muy importante ajustar lo mejor posible la altura de la fresa para que realice perfectamente la profundidad de fresado y taladrado completo. En este momento es posible empezar el fresado, manteniéndose cerca del pulsador de emergencia y sin olvidarse del mecanizado.

4.1.2. Impresión 3D

El material utilizado en la impresión es el termoplástico PLA (ácido poliláctico).

Como la carcasa de la batería no necesita propiedades estructurales importantes, se utiliza tecnología de impresión en tres dimensiones para su creación.

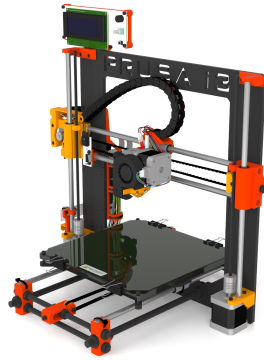


FIGURA 19. IMPRESORA 3D BQ PRUSA I3 HEPHESTOS

Estas máquinas crean piezas volumétricas a partir de diseños generados por ordenador. En este caso, la impresora es aditiva de inyección de polímero y va depositando el material por capas conforme avanza la impresión. Su funcionamiento se basa en un cabezal extrusor con tres grados de libertad en las dimensiones XYZ y un software que secciona un modelo 3D en capas para su impresión. Esta tecnología permite realizar modelos con formas complejas de manera relativamente sencilla.

Igual que con el fresado por control numérico el primer paso consiste en crear un diseño CAD.



FIGURA 20. DISEÑO CARCASA DE LA BATERÍA

En este caso al necesitar un modelo en tres dimensiones el software utilizado es Autodesk Inventor Professional 2015. Este se basa en técnicas de modelado paramétrico, que permiten modelar la geometría, dimensiones y material de las piezas. Tiene además la gran ventaja de crear ensamblajes a partir de diferentes piezas. Las piezas son ensambladas agregando restricciones lo que permite visualizar, antes de ser fabricadas, si encajarán en el diseño final o en caso contrario si no se acoplan bien y deben ser modificadas o desechadas. Es por esta cualidad que incluso los diseños en metacrilato, que no necesitan modelos en 3D, conviene realizarlos asimismo con este software. Así se pueden realizar los ensamblajes incluyendo el resto de componentes, como pueden ser en este caso los servos, para verificar que las dimensiones escogidas son las apropiadas.

Una vez finalizado el diseño de la pieza y comprobada su viabilidad se debe exportar en formato STL (STereo Litography). Este formato define la geometría de los objetos en tres dimensiones excluyendo información no relevante para el prototipo como son el color, texturas o propiedades físicas que sí incluyen otros formatos de diseño asistido por computador. Los archivos en este formato se abren, seguidamente, con el programa Slic3r. Esta herramienta convierte el modelo digital en tres dimensiones en instrucciones de impresión. Para ello secciona el modelo en capas, generando las trayectorias para rellenarlas y calculando la cantidad de material a extruir. Entre las diferentes opciones que se pueden seleccionar para esta tarea se encuentran la elección del grosor de las capas y la densidad de relleno de estas. El archivo creado por Slic3r, llamado G-code, se abre a continuación con Pronterface. Este programa es el encargado de comunicarse y manejar la impresora Prusa i3 a partir del código en lenguaje G. También dispone una interfaz gráfica (GUI) en la que el usuario puede controlar el movimiento del cabezal y las temperaturas del extrusor y de la cama de

impresión. La impresora estará lista para imprimir cuando alcance la temperatura de trabajo.

4.2. Banco de pruebas

El banco de pruebas sirve para la verificación terrestre del diseño final aéreo, comprobándose el movimiento del helicóptero mediante el desplazamiento del centro de gravedad.

Para su fabricación se utilizan piezas de metacrilato, a excepción de la pieza de la batería. Este plástico transparente en un 93% destaca principalmente por su alta resistencia al impacto y una densidad de tan sólo 1190 kg/m^3 . El metacrilato se comercializa en planchas rectangulares de diferentes grosores.

Una vez fabricadas todas las piezas, siguiendo los métodos descritos en los puntos anteriores, se procede a realizar el montaje del banco de pruebas. Las piezas de metacrilato han sido diseñadas de tal forma que puedan ser montadas sin necesidad de utilizar elementos de fijación, basta con ensamblar las pestañas en sus respectivas ranuras.

Como se puede observar en la siguiente figura, el banco de pruebas está formado por dos partes bien diferenciadas. Por un lado consta de una base, con sus dos paredes, que es la encargada de sostener al helicóptero. Las paredes están reforzadas estructuralmente con dos contrafuertes, ya que debido a su gran altura en comparación con su estrecho grosor, lo que las hace ser demasiado flexibles para la aplicación que se necesita.

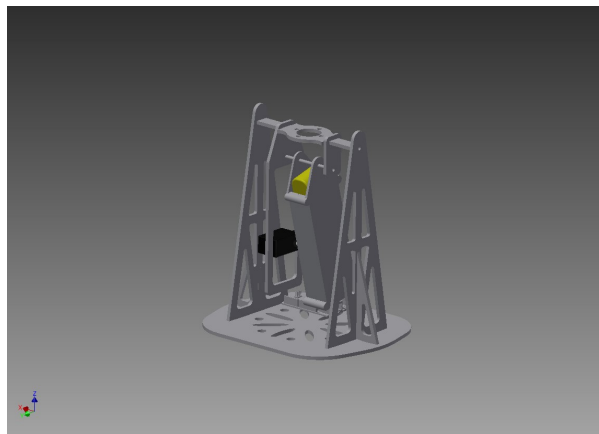


FIGURA 21. DISEÑO PRIMER BANCO DE PRUEBAS

La función principal de la base es la de permitir el giro del helicóptero con el desplazamiento del centro de gravedad pero impidiendo que pueda volcar o despegar. Al añadir ruedas a la base se permite a su vez que esta pueda desplazarse por el suelo. La segunda parte del banco de pruebas es la relativa al helicóptero. Este primer prototipo simplificado cuenta sólo con los motores, la batería y un servo para desplazar esta batería. Esta configuración permite verificar el correcto funcionamiento de los rotores y conocer el giro que realiza el motor como reacción al movimiento de la batería. Como no se conoce el lugar óptimo para colocar el servomotor, existe un raíl con diversos anclajes en el que puede ir cambiando su posición. El eje que permite el movimiento entre la base y el helicóptero está formado por unos alfileres encajados en unos taladros realizados longitudinalmente en el metacrilato.

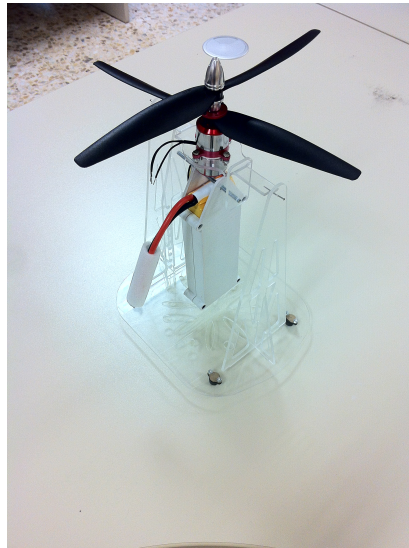


FIGURA 22. FOTOGRAFÍA BANCO DE PRUEBAS

En las primeras pruebas realizadas se encuentran diversos defectos en el diseño de la estructura, entre ellos la falta general de rigidez. Otro punto objeto de mejora es el eje que permite la rotación del helicóptero. Aparte de su debilidad, este se encuentra alejado del centro de gravedad de la futura aeronave. Este aspecto impide que el movimiento observable en el banco de pruebas se asemeje al del futuro helicóptero en vuelo. Por ello se decide realizar un segundo diseño.

En la siguiente figura se contempla como se ha desplazado el eje sobre el centro de gravedad. Además se ha reforzado la estructura del helicóptero y sustituido el eje de alfileres por unos tornillos de métrica M4. Al descender el eje de giro se ha

podido recortar la altura de las paredes de la base consiguiendo así que también aumente su robustez. Por último, se aprovecha la creación de un nuevo prototipo para añadir sujeciones en las que colocar la placa microcontroladora y los controladores electrónicos de los motores.

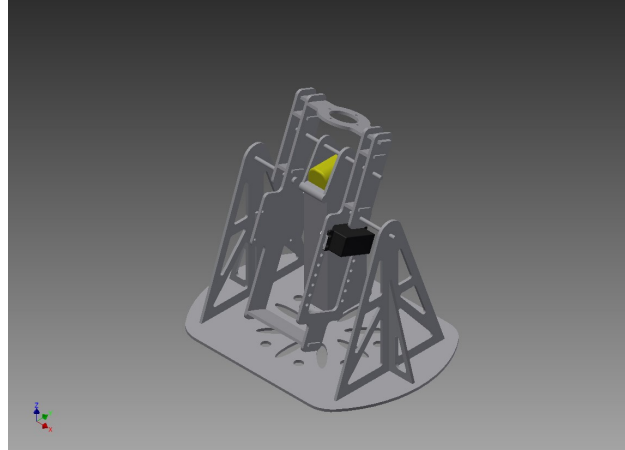


FIGURA 23. SEGUNDO BANCO DE PRUEBAS

4.3. Conexión de los componentes

Tanto los variadores electrónicos, como la batería y los motores comprados no disponen de conectores para realizar las conexiones. Por ello es necesario elegir unos conectores apropiados y soldarlos. En este caso se necesitan dos conectores de oro de 6 *mm* para la alimentación de la batería y seis conectores de tipo banana de 2 *mm* para las señales trifásicas de ambos motores. La batería se conecta en la entrada de los dos controladores electrónicos y los tres cables de salida de estos variadores se conectan a cada uno de los motores.

Debido a la falta de marcas o señales a la hora de realizar las conexiones, no se conoce que variador controla cada uno de los motores. Para averiguarlo hay que descubrirlo experimentalmente. Otro de los fenómenos que se pueden dar es a la hora de arrancar los motores es si alguno de ellos o los dos giran en sentidos equivocados. En este caso, para solucionarlo, basta con intercambiar entre si dos de las tres fases que transmiten la señal de los controladores a los motores.

El siguiente paso es conectar los conectores de servo, tanto de los variadores como del servomotor, a la placa Arduino. Para facilitar esta tarea se utiliza una placa de prototipado.

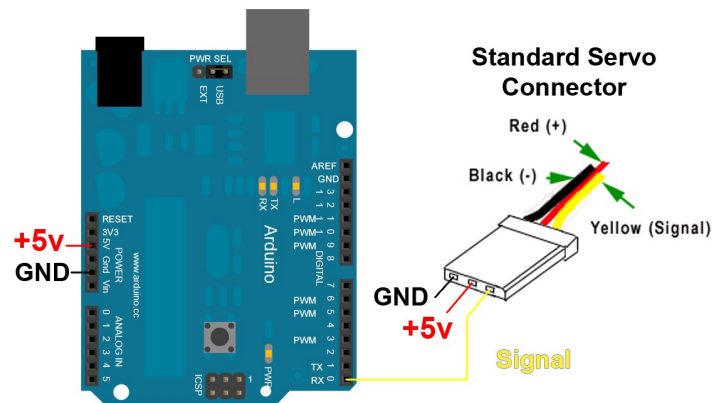


FIGURA 24. CONEXIÓN DE CONECTOR TIPO SERVO

Los conectores de tipo servo tienen tres cables. El de color rojo es para la alimentación positiva, siendo lo habitual 5V, el negro para la referencia de tierra y el último de color naranja o amarillo es el de señal. En el caso de trabajar con una placa Arduino hay que asegurarse que el cable de señal se conecta a uno de los pines digitales preparados para mandar señales por modulación por ancho de pulso.

A continuación, se muestra un esquema con las conexiones, en el diseño final, de todos los componentes del helicóptero.

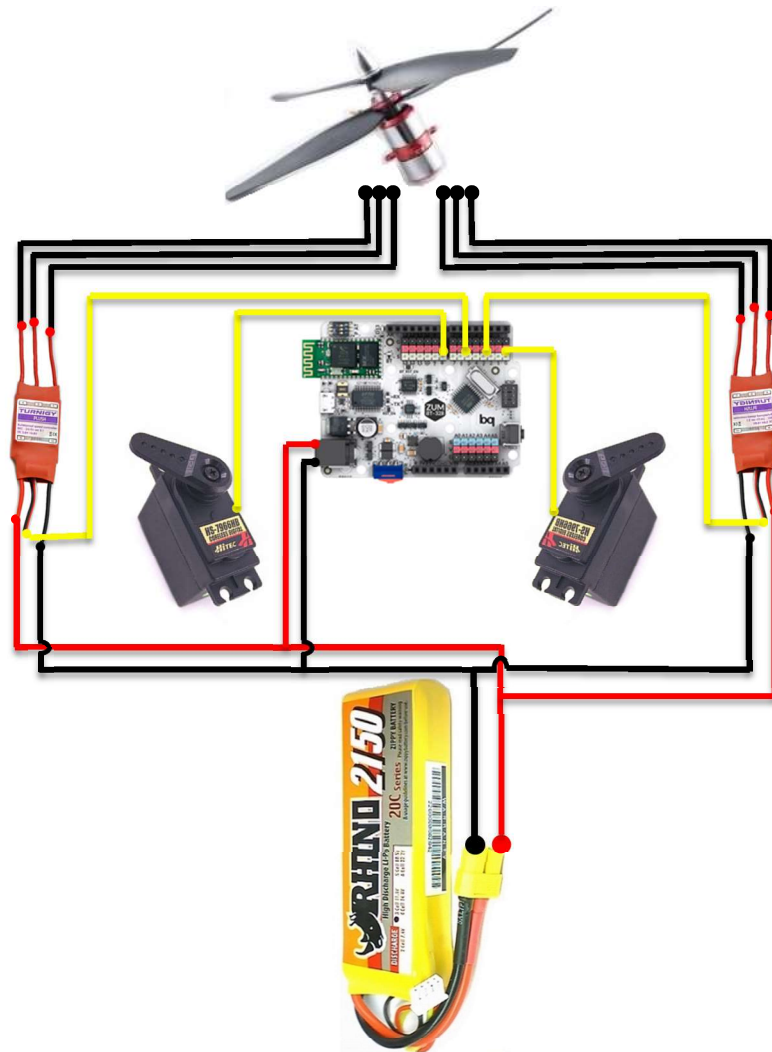


FIGURA 25. ESQUEMA DE CONEXIONES

4.4. Arduino Yún

En un primer momento, se elige la placa Arduino Yún para facilitar la comunicación con el teléfono.

Esta placa combina Arduino con una distribución de Linux llamada Linino, específica para Arduino. Por un lado se integra un sistema Arduino basado en el procesador Atmega32U4 y por otro un procesador con una distribución basada en OpenWRT con conectividad WiFi y con Linino instalado. Dispone también de los pines necesarios de entrada y salida.

TABLA 11. ESPECIFICACIONES TÉCNICAS ARDUINO YÚN

Microcontrolador:	ATmega32u4
Voltaje entrada:	5V
Pines digitales E/S:	20
Pines digitales E/S PWM:	7
Pines analógicos entrada:	12
Frecuencia reloj:	16 MHz
Memoria Flash:	32 kB
WiFi:	IEEE 802.11 b/g/n
Dimensiones:	53 x 73 mm
Peso:	32 g

Una particularidad de esta placa es la librería Bridge que facilita la comunicación entre los dos procesadores permitiendo que los programas de Arduino ejecuten comandos en Linino y accedan a los periféricos conectados al procesador AR 9331.

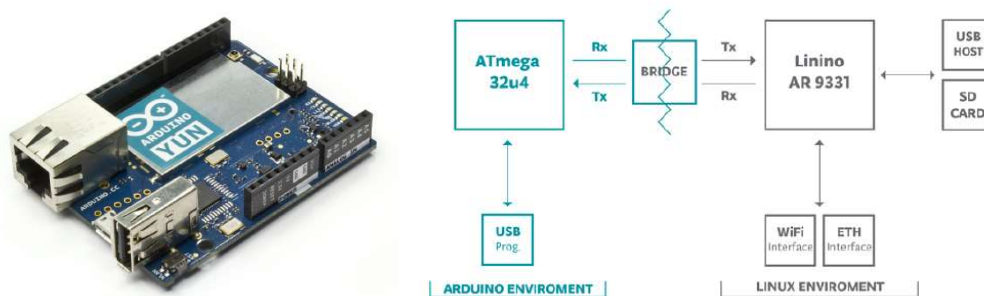


FIGURA 26. ARDUINO YÚN Y LIBRERÍA BRIDGE

Para el desarrollo del código de esta placa, se utiliza el programa Arduino IDE. Con la programación de un *sketch*, se pretende manejar el servo y los motores a través de la tecnología WiFi, puesta a disposición en esta placa. Esto evita la necesidad de mantener el controlador conectado directamente a un ordenador para recibir los comandos de movimiento. Como consecuencia, al ser inalámbrico, permite rodar al banco de pruebas por el suelo sin verse afectado por el arrastre de cables.

Este *sketch* crea un servidor HTTP en el microcontrolador. Este programa es el encargado de abrir un puerto de comunicación con los clientes de la aplicación y atender sus peticiones. Para que el cliente, en este caso el operario del

helicóptero, pueda enviar comandos de manera sencilla se crea una interfaz gráfica con HTML.

COAX Controller v1.0

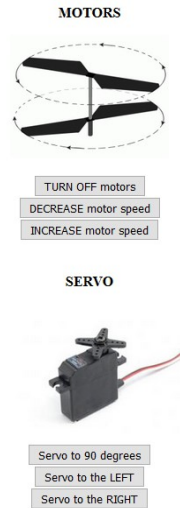


FIGURA 27. INTERFAZ GRÁFICA EN HTML

Así, el cliente, solo tiene que acceder desde su navegador a la dirección <http://yun.local/arduino/>, mientras está conectado a la misma red local que la placa Yún, para poder manejar el helicóptero.

4.5. Placa PCB

Durante las pruebas se detecta el inconveniente de realizar las conexiones de los terminales servo en la placa de prototipado. El elevado número de cables necesario para realizar todas las conexiones hace engorrosa la tarea y aumenta la probabilidad que alguno de ellos pueda soltarse o sea colocado incorrectamente, además del tamaño y peso añadido. Por ello se decide diseñar una placa de circuito impreso que facilite dicha tarea.

Para el diseño de la placa se utiliza el software EAGLE de la compañía CadSoft Computer. Con él se pueden trazar las pistas de material conductor necesarias para conectar eléctricamente y de manera apropiada las diferentes partes del circuito. En este caso el diseño resulta muy sencillo debido a la ausencia de componentes electrónicos utilizados. Basta con delinear los caminos para

conectar cada uno de los cuatro conectores tipo servo necesarios a la alimentación, tierra y sus respectivas señales. Para que la placa resulte compacta y fácil de instalar en el helicóptero se diseña de forma que pueda ser montada sobre la placa Arduino. Este tipo de placas externas se conocen como *shield* y se introducen en los pines de entrada y salida de la placa Arduino. Para no obstruir el acceso a los pines, se extienden éstos a un nivel superior. El diseño terminado se puede visualizar con la herramienta Gerbv. Este programa permite comprobar el correcto trazado de las pistas así como la colocación de los agujeros y el perfil escogido para la placa. A continuación se fresa la placa y se sueldan los diferentes conectores. Se necesitan cuatro regletas hembra de 2,54 mm para extender los pines de la placa Arduino, cuatro pines triples macho de 2,54 mm para los conectores tipo servo y una regleta clema con dos contactos para conectar la alimentación.

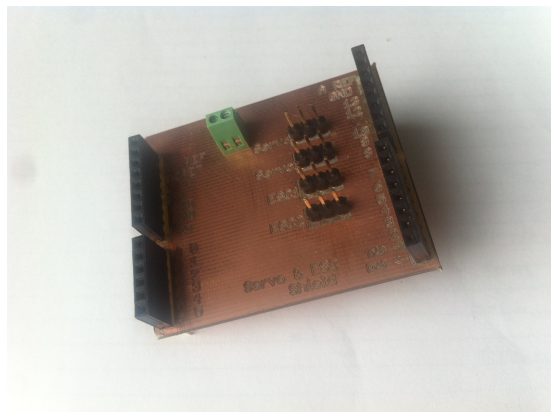


FIGURA 28. PLACA PCB

4.6. Placa BQ ZUM BT-328

Después de la fabricación de la placa *shield*, detallada en el punto anterior, surge la posibilidad de cambiar la placa Arduino Yún por una Placa ZUM BT-328 de la empresa BQ. Con la utilización de esta placa disponemos de pines macho para el conexionado directo de conectores servo, descartando así la placa PCB.

El cambio de la tecnología WiFi por Bluetooth no resulta negativo, ya que sólo es necesaria para la comunicación entre la placa y el teléfono móvil montado en el helicóptero. Estos se encuentran a una distancia de unos centímetros, no siendo determinante la distancia mínima alcanzada por el Bluetooth. Por estos

motivos, además de las cualidades de la placa ZUM BT-328, como puede ser el mayor rango de voltaje de alimentación, se descarta el uso de la placa Arduino Yún en favor de la placa ZUM.

El microcontrolador está provisto de un módulo Bluetooth, que viene con una configuración de fábrica, pero su reconfiguración está disponible con la documentación del fabricante, Bolutek (15). Siguiendo las instrucciones, a través de comandos AT, se cambia la tasa de baudios a 115200 bps, que es la máxima velocidad que permite el Serial de Arduino IDE, dejando este valor constante para todo el desarrollo del proyecto. La tasa de baudios que viene por defecto en el módulo es de 9600 bps.

Para la realización de pruebas de comunicación inalámbrica, se descarga la aplicación, sin coste, ArduinoRC para teléfonos con sistema operativo Android.

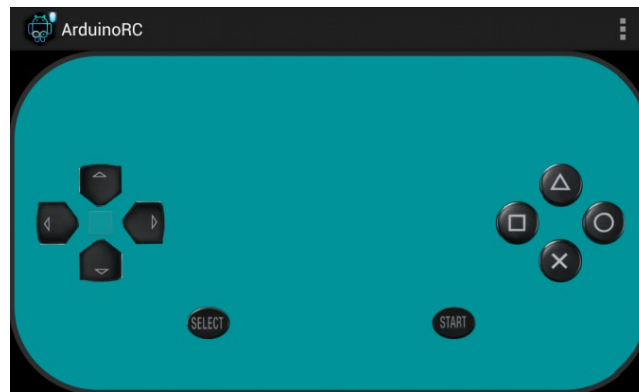


FIGURA 29. INTERFAZ DE LA APLICACIÓN ARDUINORC

Esta aplicación permite, después de emparejar el teléfono con la placa ZUM, mandar comandos a través de Bluetooth. Además tiene una interfaz gráfica con los controles necesarios para que el manejo del helicóptero sea más intuitivo.

El desarrollo de los programas Arduino están recogidos en la memoria de mi compañero, Eduardo Ortega (1).

4.7. Configuración, manejo y control

A la hora de desarrollar el diseño de una aeronave de ala giratoria, aparte del número de rotores y su configuración, es necesario elegir el mecanismo para su manejo. En los helicópteros convencionales, el movimiento se consigue variando el ángulo de ataque de las palas del rotor, que según su posición, se regula la sustentación creada por la pala. Esta característica, aunque eficaz, hace compleja su fabricación debido al elevado número de piezas necesarias para elaborar este mecanismo.

En este trabajo se diseña un mecanismo capaz de mover el centro de gravedad de la aeronave que provoque que esta se incline junto con sus hélices. Esta inclinación varía el ángulo de empuje de los rotores y, a su vez, origina el desplazamiento del helicóptero. En el siguiente punto se explica más detalladamente el mecanismo para variar el centro de gravedad.

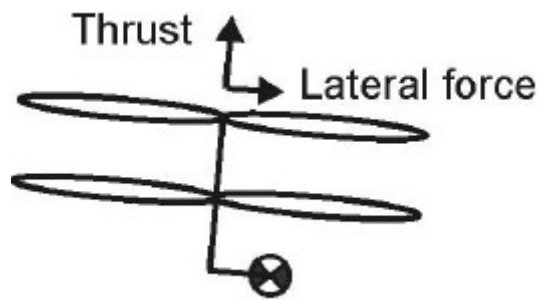


FIGURA 30. DESPLAZAMIENTO DEL HELICÓPTERO VARIANDO SU CENTRO DE GRAVEDAD

El control de actitud del helicóptero se ejecuta en el Smartphone a bordo. Con los sensores del móvil, se consigue actualizar la orientación del terminal, en consecuencia la del helicóptero, y enviar consignas para el control de la aeronave.

Para el control de altura se utiliza el sensor ultrasónico, explicado con más detalle en el informe de Eduardo Ortega (1).

4.8. Pruebas y experimentos

En este apartado, se describen las diferentes pruebas que se han llevado a cabo durante el desarrollo del helicóptero.

A medida que la estructura va cambiando de forma se prueba experimentalmente para poder verificarla y obtener las mejoras necesarias en la siguiente evolución.

La primera verificación de los componentes se efectúa con el motor anclado a una madera rígida para evitar un posible despegue.



FIGURA 31. PRUEBA COMPONENTES

De este ensayo, se confirma el buen funcionamiento de los diferentes componentes: motores, controladores electrónicos, batería y microcontrolador.

Una vez montado el primer banco de pruebas, se realiza la prueba de movimiento del sistema mediante el desplazamiento del centro de gravedad.

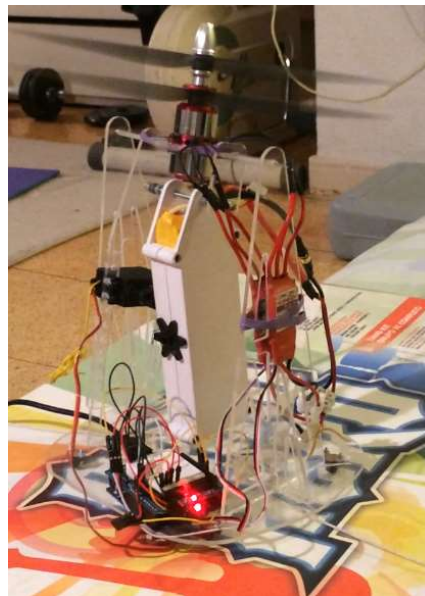


FIGURA 32. MOVIMIENTO DEL SISTEMA MEDIANTE EL DESPLAZAMIENTO DEL CENTRO DE GRAVEDAD

Esta prueba resulta exitosa, verificando el movimiento del helicóptero. Tras este ensayo se continúa con la fabricación del segundo banco de pruebas y los diseños posteriores.

Con la construcción de los diseños finales del helicóptero, se decide montar una jaula para la realización de pruebas de vuelo. Esta jaula ha sido diseñada con la finalidad de poder realizar experimentos con total seguridad, al crear una barrera física entre la aeronave y las personas que la operan.

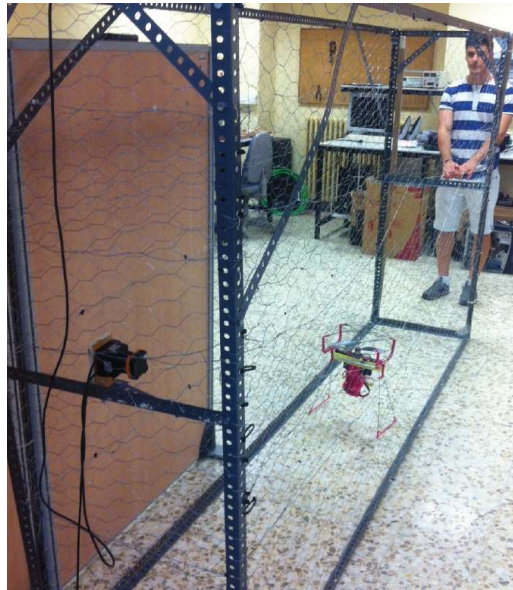


FIGURA 33. ENSAYO DE VUELO EN LA JAULA

Como se observa en la figura anterior, se implementa también una medida de seguridad manual que consiste en el paso de un hilo por unas guías, puestas manualmente en el helicóptero, para producir la detención del helicóptero al realizar una tensión completa del hilo, evitando caídas o golpes por el descontrol de la aeronave.

En esta jaula se prueban experimentalmente los diferentes prototipos del helicóptero con la finalidad de observar su comportamiento en vuelo. Del mismo modo, también se registra el movimiento mediante el láser Hokuyo UTM-30LX y se prueba el control de altura del helicóptero con el sensor ultrasónico. Estos últimos ensayos, están recogidos en la memoria de mi compañero, Eduardo Ortega (1).

Capítulo 5

Diseño Final

En este capítulo, se muestra el diseño final del helicóptero. El desarrollo de los distintos diseños de los prototipos intermedios hasta llegar al diseño final, se muestran en el trabajo de Eduardo Ortega (1).



FIGURA 34. DISEÑO FINAL HELICÓPTERO COAXIAL

Este diseño presenta, con su estructura híbrida de metacrilato y plástico PLA, el esqueleto con mayor robustez. El soporte para el motor es la única pieza fresada en metacrilato, este material posee la rigidez necesaria para servir de base al resto del vehículo. Además absorbe las vibraciones de los motores sin provocar un exceso de fatiga en la estructura.

La protección, con varillas de fibra de carbono en forma circular, ha resultado ser eficaz en las pruebas realizadas. Son capaces de absorber golpes y caídas, y protegen al entorno de las hélices. Su facilidad de montaje es considerada como ventajosa. Si esta pieza se rompe en un accidente se puede volver a imprimir y colocar de nuevo en la aeronave fácilmente.

El cuerpo del helicóptero está formado por el mecanismo empleado para desplazar el centro de gravedad y el soporte para el Smartphone.

El primero se diseña para ser capaz de mover el centro de gravedad, lo conforma la batería con dos grados de libertad para que la aeronave pueda realizar las maniobras de alabeo y balanceo. Esto se consigue mediante el movimiento de los servomotores que posicionan, con el mecanismo, la batería en la situación deseada.

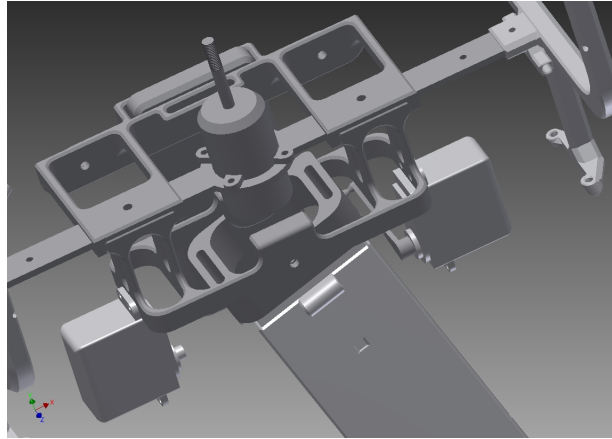


FIGURA 35. MECANISMO PARA DESPLAZAR EL CENTRO DE GRAVEDAD

El primer servomotor mueve la pieza donde cuelga la batería. El movimiento es de “derecha a izquierda”, siguiendo la dirección del eje principal donde está colocado el motor. Es en el segundo, donde se utiliza la junta cardan para mover la pieza donde se encuentra la batería. La junta cardan es un componente mecánico, que permite unir dos ejes no colineales. Mediante el uso de este elemento, es posible el movimiento de “adelante y atrás”, perpendicular al eje principal, respetando los movimientos del otro servomotor.



FIGURA 36. JUNTA CARDAN

El soporte para el teléfono móvil es estable y con una buena fijación al eje de los motores. El Smartphone, al realizar los cálculos de control del helicóptero a través de sus propios sensores, deberá mantenerse lo más sujeto posible a la estructura, evitando las posibles perturbaciones que pueda haber.

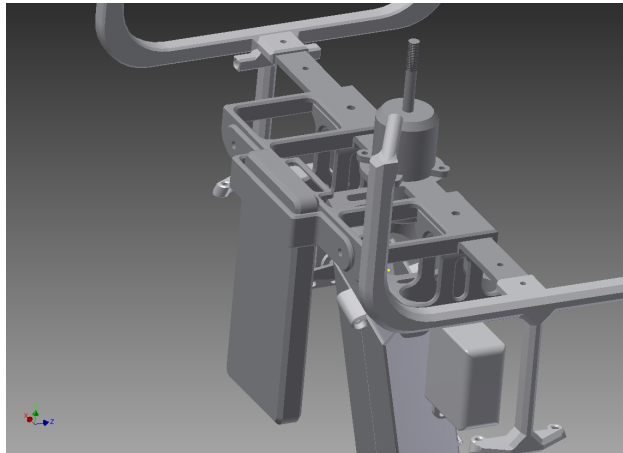


FIGURA 37. SOPORTE DEL TELÉFONO MÓVIL

En la siguiente imagen se muestra el diseño final del helicóptero.

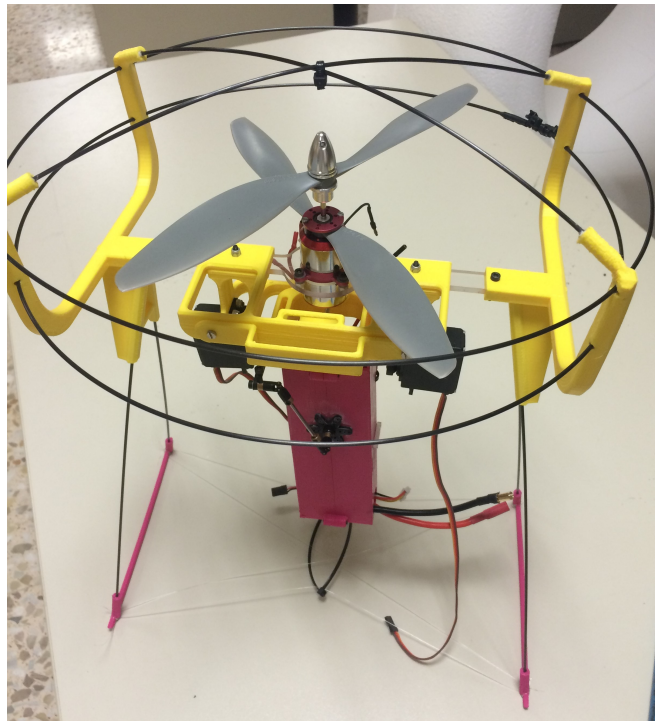


FIGURA 38. FOTOGRAFÍA DISEÑO FINAL HELICÓPTERO

Capítulo 6

Aplicaciones Android

En este capítulo se describen las aplicaciones desarrolladas para ser ejecutadas en los Smartphones con el sistema operativo Android. En primer lugar, se describe brevemente el sistema operativo, el entorno de desarrollo y la comunicación, para luego, a continuación, explicar por separado cada una de las aplicaciones, BTController y OnBoard Control.

6.1. Android

Android es un sistema operativo basado en el núcleo Linux. Fue diseñado principalmente para dispositivos móviles con pantalla táctil. Inicialmente fue desarrollado por Android Inc., empresa que Google respaldó económicamente y más tarde, en 2005, compró. Android fue presentado en 2007 junto la fundación del Open Handset Alliance (un consorcio de compañías de hardware, software y telecomunicaciones) para avanzar en los estándares abiertos de los dispositivos móviles.

Es por esto último, por lo que se decide utilizar este sistema operativo. El gran volumen de dispositivos con este sistema, con una enorme fuente de documentación pública y código abierto, hacen de Android el sistema idóneo para desarrollar las aplicaciones que se implementarán en el helicóptero.



FIGURA 39. LOGOTIPO ANDROID

6.2. Android Studio

Android Studio es un entorno de desarrollo integrado para la plataforma Android. Fue anunciado el 16 de mayo de 2013 en la conferencia Google I/O, y reemplazó a Eclipse como el IDE oficial para el desarrollo de aplicaciones para Android. La primera versión estable fue publicada en diciembre de 2014.



FIGURA 40. ICONO ANDROID STUDIO

El primer punto a tener en cuenta antes de iniciar el desarrollo de las aplicaciones, es la elección del entorno de desarrollo. Se elige Android Studio como herramienta por ser, al principio de la realización de este trabajo, el IDE oficial para el desarrollo de aplicaciones para Android y, por lo tanto, el que da soporte y actualizaciones a las versiones más recientes del firmware.

6.3. Comunicación

En este punto se detallan las comunicaciones llevadas a cabo entre los dispositivos móviles que comprenden el helicóptero. Por orden, se explican las tecnologías de comunicación inalámbricas, se expone un diagrama con las direcciones de comunicación de cada elemento y, por último, se detalla el protocolo de comunicación empleado.

6.3.1. Tecnologías inalámbricas

Para la comunicación entre los Smartphones, se estudian las dos siguientes dos opciones inalámbricas posibles.

La primera es Bluetooth, que es una especificación industrial para Redes Inalámbricas de Área Personal (WPAN), que posibilita la transmisión de datos

entre diferentes dispositivos mediante enlace por radiofrecuencia. Está diseñado especialmente para dispositivos de bajo consumo y que requieran corto alcance de emisión. El alcance varía, dependiendo de la potencia de transmisión, de 10 m para potencias de 2,5 mW (Clase 2) a 100 m para potencias de 100 mW (Clase 1).

La segunda tecnología, también basada en la transmisión de datos inalámbrica por radiofrecuencia, es la conocida como WiFi o su estándar IEEE 802.11. Tiene un alcance de 200 m y una velocidad máxima de 11 Mbps en el estándar 802.11b, 100 m y 54 Mbps en el estándar 802.11g y 250 m de alcance a una velocidad de 108 Mbps en la última versión 802.11n.

A pesar de que todo apunta al uso de la segunda tecnología como forma de comunicación inalámbrica entre los teléfonos inteligentes, por su mayor alcance, surge el problema de que los teléfonos disponibles no soportan el firmware Android 4.0 (API 14). Esto conlleva que no pueden lanzar su propio punto de acceso local WiFi, ya que los métodos de Wi-Fi Peer-to-Peer de Android solo están disponibles para el software antes mencionado y versiones posteriores.

De este modo, la tecnología utilizada para la comunicación entre los Smartphones es Bluetooth, que, aunque tiene un menor alcance, nos servirá para validar el comportamiento y funcionamiento del helicóptero. Así mismo, se intenta programar las aplicaciones de una forma modular para una fácil actualización a otro tipo de tecnología de comunicación, para lograr el control a mayor alcance, como puede ser WiFi o la red de datos 4G.

6.3.2. Diagrama global

Con este diagrama se pretende, a modo esquemático, mostrar las comunicaciones que se producen en todo el sistema. En los puntos posteriores de cada aplicación se detallan todos los procesos expuestos en la siguiente figura.

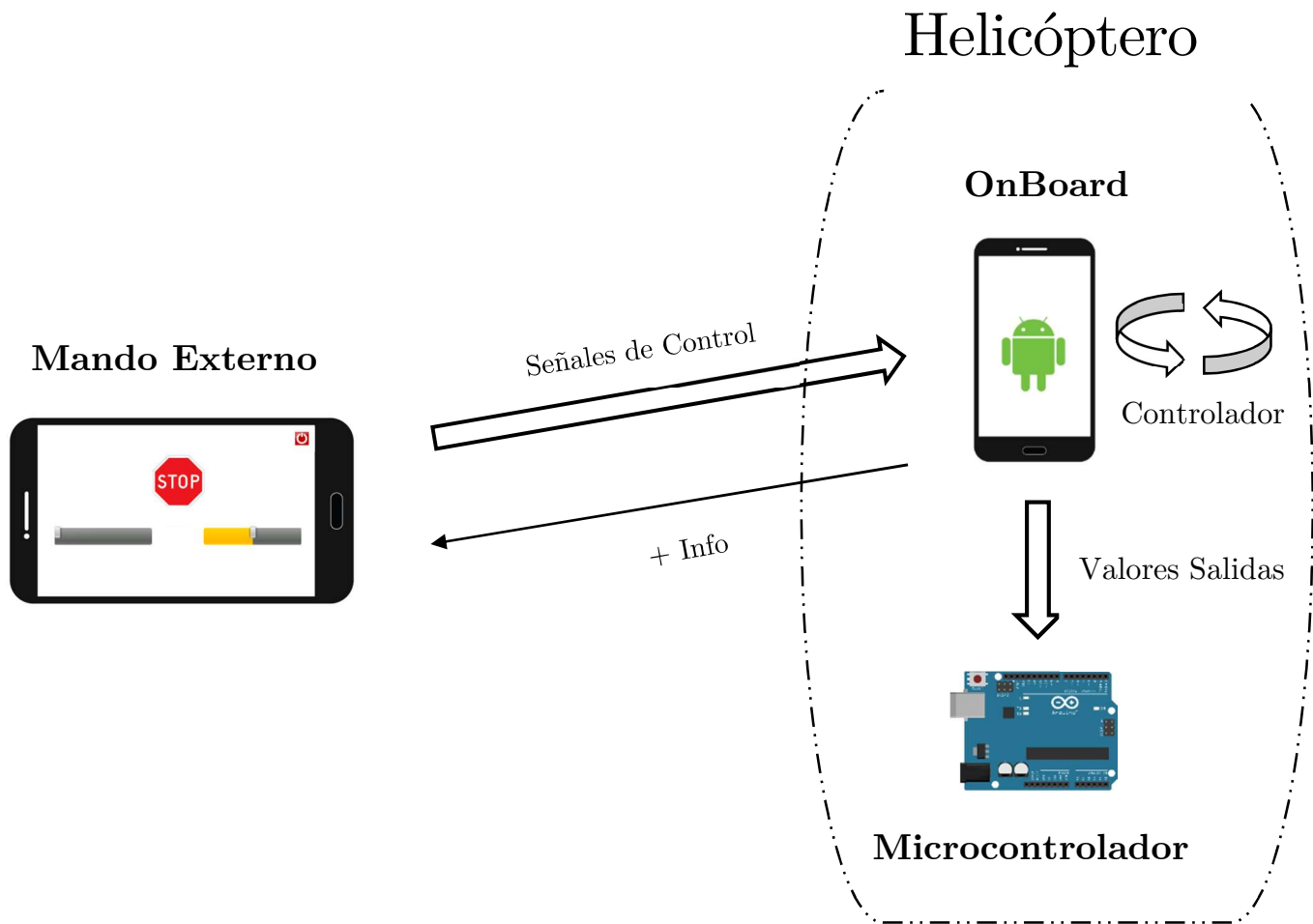


FIGURA 41. DIAGRAMA COMUNICACIONES

6.3.3. Protocolo

Es importante normalizar un protocolo de comunicación, que establezca las reglas para enviar la información a través de Bluetooth. Esta guía permite que los dispositivos sean capaces de interpretar los datos recibidos, detectar y alertar si ha habido errores en la comunicación. Para ello se definen la sintaxis, semántica y sincronización de los conjuntos de datos o paquetes de la comunicación.

Teniendo en cuenta esto último, se fija una cabecera que precede al envío de cada mensaje. Como la comunicación serie Bluetooth se realiza mediante el envío de bytes en valores decimales comprendidos entre el 0 y el 255, se decide que el byte con valor 255 ejerza de cabecera. Por lo tanto, se entiende que al recibir dicho byte da comienzo la recepción de un nuevo mensaje. Esta cualidad implica que el byte 255 debe estar exclusivamente reservado para indicar el comienzo de un

mensaje, no pudiendo tomar ningún otro byte del mensaje dicho valor. Al incluir los rangos binarios entre 0 y 250, se evita la posibilidad de que se transmita una secuencia de ocho bits a uno, fortaleciendo el papel del byte de cabecera.

Después del byte de cabecera, se envían los bytes con las señales que se desean transmitir de un dispositivo a otro. Estos bytes tienen diferente significado si el envío es del mando exterior al móvil OnBoard o es del OnBoard al microcontrolador. Por lo tanto, se explicarán con mayor detalle más adelante, en el punto correspondiente de cada aplicación.

Por último, se añade un código de detección de errores conocido como verificación por redundancia cíclica (CRC por sus siglas en inglés). Este código se utiliza, con frecuencia, en redes digitales y dispositivos de almacenamiento para detectar cambios accidentales en los datos. El valor de verificación utilizado, en este caso, se corresponde al resto resultante de la división, en la que tomamos como dividendo la suma de los bytes de información y como divisor el dígito 256.

$$CRC = \left(\sum \text{Bytes del mensaje} \right) \% 256$$

El empleo de este código consiste en calcular el valor CRC en el emisor y enviarlo como último byte del mensaje. Por su parte el receptor, al recibir el mensaje completo, vuelve a calcular este valor y comprueba si coincide con el CRC recibido. De esta forma, si cualquiera de los bytes recibidos varía con el valor de los bytes enviados, el código de detección detecta la anomalía en el valor de verificación y desecha el mensaje.

Al final se concluye que cada paquete está formado por un byte de cabecera, cuatro bytes de mensaje y un último byte de verificación.

Cabecera	Byte 1	Byte 2	Byte 3	Byte 4	CRC
255	0 - 250	0 - 250	0 - 250	0 - 250	(Byte (1 + 2 + 3 + 4)) % 256

FIGURA 42. COMPOSICIÓN DE UN MENSAJE DEL PROTOCOLO

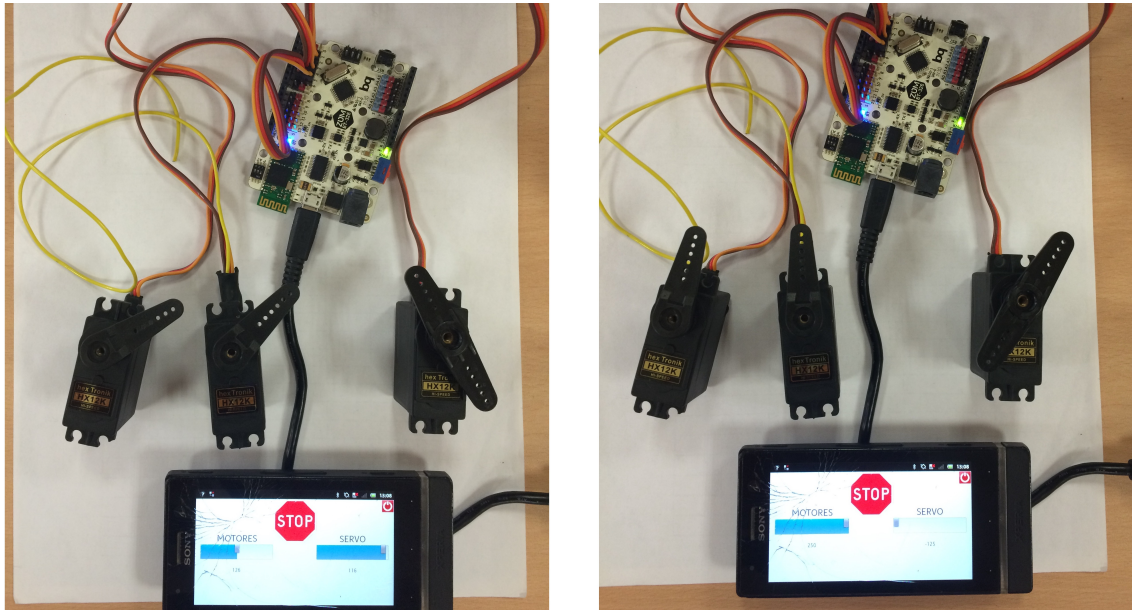


FIGURA 43. PRUEBA PROTOCOLO DE COMUNICACIÓN MEDIANTE BLUETOOTH

6.4. Test de velocidad

Antes de comenzar a diseñar las aplicaciones, se decide testar la velocidad de envío Bluetooth del terminal. La velocidad de los mensajes es una de las actividades críticas en el sector del aeromodelismo; por ello, es conveniente examinar la velocidad de esta tecnología inalámbrica, para conocer la escala de tiempo aproximada de envío.

Para que esto sea posible, se necesita un receptor que devuelva el mensaje enviado, utilizando la placa ZUM BT-328 para este destino. Se programa un pequeño *sketch* para la recepción del carácter enviado desde el Smartphone e inmediatamente lo devuelve en sentido contrario.

Por la parte del Smartphone, se desarrolla una aplicación Android para que envíe un carácter o una cadena de texto, por Bluetooth, al microcontrolador, y seguidamente, permanezca a la escucha.

El tiempo actual en nanosegundos se puede conocer mediante la función *System.nanoTime()*. El tiempo de envío se calcula hallando la mitad del resto de la operación, en la que tomaremos como minuyendo el tiempo de llegada del mensaje y como sustrayendo el tiempo en el que se envió este.

```

long tiempo_envio;

long t1 = System.nanoTime();
sendData(txtenviar);
beginListenForData();
long t2 = System.nanoTime();

tiempo_envio = (t2 - t1)/2;

```

FIGURA 44. MEDIDA TIEMPO DE ENVÍO

Se implementa un bucle *for* para ejecutar la sentencia anterior con las iteraciones deseadas, para posteriormente mostrar por pantalla la media del tiempo de envío en todas las medidas, en nanosegundos y milisegundos.

La posibilidad de registrar los valores de la variable del tiempo de envío en un archivo *.txt*, nos permite analizar los datos.

A continuación se muestra la gráfica del tiempo de envío en milisegundos para cien iteraciones.

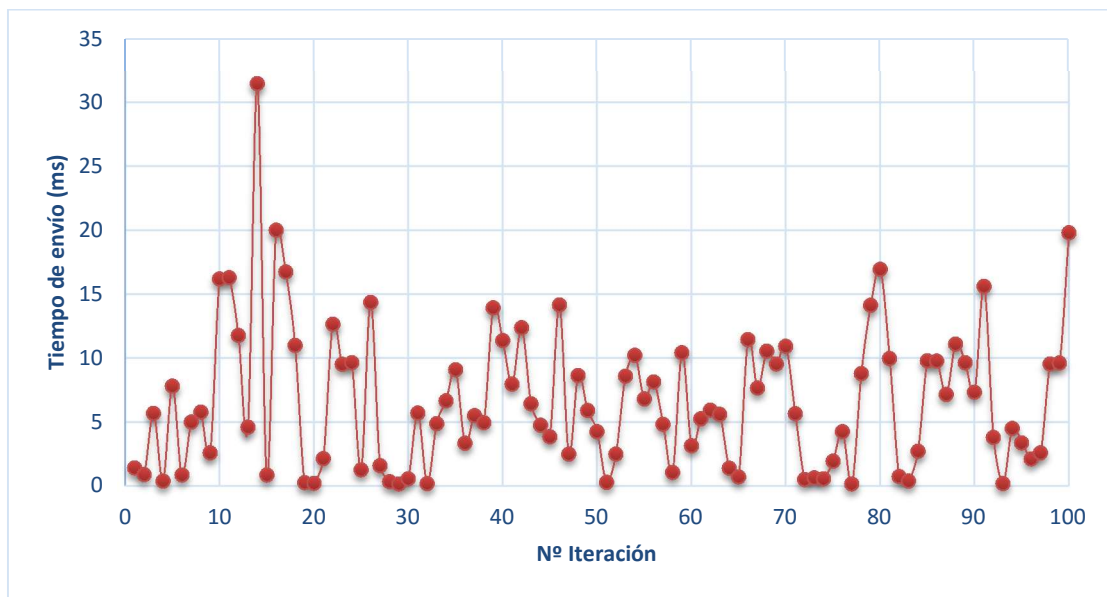


FIGURA 45. GRÁFICA TIEMPO DE ENVÍO

La reflexión, a la vista de estos datos, es que son muy poco concluyentes para poder analizarlos fehacientemente. Cuando el envío es de un carácter y una sola iteración, a priori debiera ser más preciso, las medidas son también bastante aleatorias, oscilando de medidas del orden de nanosegundos a medidas del orden de decenas de milisegundos.

Esto nos hace pensar que posibles interferencias en la comunicación producen el retraso de algunos mensajes, o que las funciones de envío y llegada producen cualquier tipo de retardo al ejecutarse en el mismo hilo de ejecución.

Este problema parece solventarse, en las aplicaciones posteriores, cuando la comunicación es principalmente unidireccional y mediante el protocolo de comunicación, no apreciándose retardo en el desarrollo.

6.5. BTController

BTController es el nombre elegido para la aplicación que ejecuta el Smartphone que actúa como mando inalámbrico.

A modo de resumen, esta aplicación es la encargada de enviar y recibir señales al teléfono que está a bordo del helicóptero. El usuario manda las consignas que desea obtener en el comportamiento de la aeronave, siendo estas la altura, yaw, pitch y roll.

BT Controller

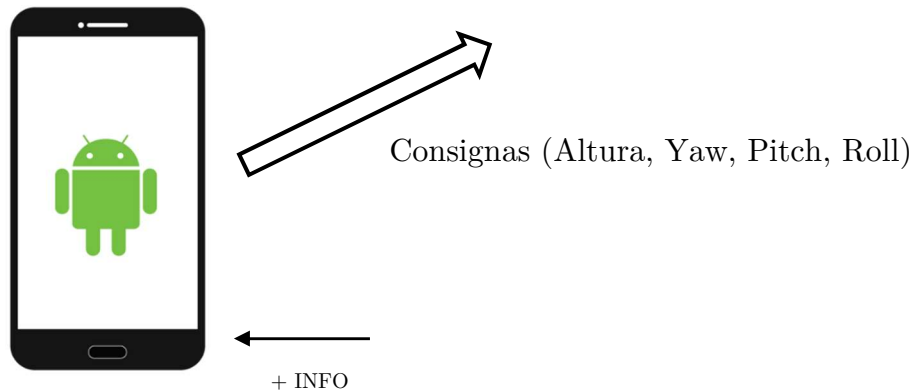


FIGURA 46. DIAGRAMA BTCONTROLLER

Al iniciar la aplicación, nos encontramos con la primera interfaz, que es dónde seleccionamos manualmente el dispositivo que está a bordo para empezar la comunicación.



FIGURA 47. PRIMERA ACTIVIDAD BTCONTROLLER

En la parte baja de la pantalla, se localiza un objeto *ListView* con todos los dispositivos vinculados por Bluetooth con el dispositivo. Se implementa así para no tener que realizar una búsqueda Bluetooth de dispositivos cada vez que se inicia la APP, solo mostrando los dispositivos vinculados con el terminal previamente.

Una vez seleccionado el móvil OnBoard, se intenta realizar la conexión y pasar a la siguiente actividad. La conexión será exitosa si el otro terminal está en modo servidor y en escucha de clientes.

Realizada la conexión, la aplicación pasa a la siguiente actividad, donde se realiza el envío de las señales.



FIGURA 48. ACTIVIDAD DE CONTROL BTCONTROLLER

Como se puede observar en la ilustración anterior, la interfaz está formada por dos *SeekBar*s, dos botones y dos cuadros de texto de información.

El botón *Stop* envía las señales de reposo al helicóptero y el botón *Logout*, arriba a la derecha, cierra el socket Bluetooth y finaliza la conexión.

Los cuadros de texto muestran la información en tiempo real del comportamiento de los mensajes. Por un lado, el cuadro situado arriba a la izquierda muestra los bytes que se están enviando de cada una de las consignas. El de la parte inferior puede presentar la altura, pitch y roll del helicóptero, siendo esta información opcional o para mostrarla en vuelos para realizar depuraciones.

Para el envío de las señales de control se utilizan los dos *SeekBar* y los datos del acelerómetro del Smartphone. Con los primeros se modifican los valores de la Altura y Yaw, desplazando la barra hacia los valores deseados, siendo estos bytes el rango del protocolo de comunicación, entre 0 y 250.

Con los datos del acelerómetro se escriben los valores de Pitch y Roll. Con la inclinación del teléfono móvil, se modifican los valores del acelerómetro y los utilizamos para mapearlos, en x e y , obteniendo las salidas.

```
@Override
public void onSensorChanged(SensorEvent event) {
    Sensor mySensor = event.sensor;
    float x=0, y=0;

    if (mySensor.getType() == Sensor.TYPE_ACCELEROMETER) {
        x = event.values[0];
        y = event.values[1];
    }

    progressPitch = (int)map(x,minPitch,maxPitch,BYTE_MIN,BYTE_MAX);
    progressRoll = (int)map(y, minRoll, maxRoll, BYTE_MIN, BYTE_MAX);
}
```

FIGURA 49. CÓDIGO LECTURA ACELERÓMETRO

El acelerómetro del Smartphone tiene los ejes de coordenadas de la siguiente imagen. Esto significa que, cuando el móvil está en una mesa plana horizontal, el valor de la aceleración en z es $9,81 \text{ m/s}^2$, el valor de la gravedad en la tierra (16).

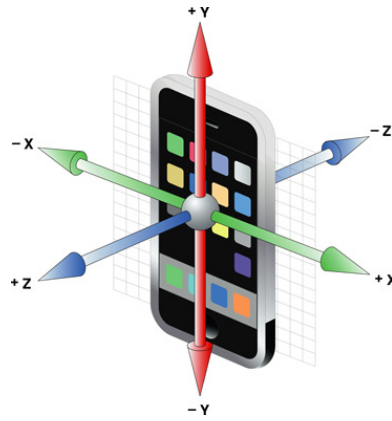


FIGURA 50. EJES DE COORDENADAS SMARTPHONE

De este modo, con el ejemplo anterior, se deduce que el cero, para los valores del acelerómetro en x e y , se obtiene cuando el terminal está en posición horizontal con la pantalla hacia arriba.

Por lo tanto, mediante una inclinación manual del Smartphone, de “adelante y atrás”, variamos el Pitch, con el ángulo ajustado, obteniendo valores en el rango del protocolo de comunicación.

De la misma manera, una inclinación manual de “derecha e izquierda” varía el Roll, obteniendo valores entre 0 y 250.

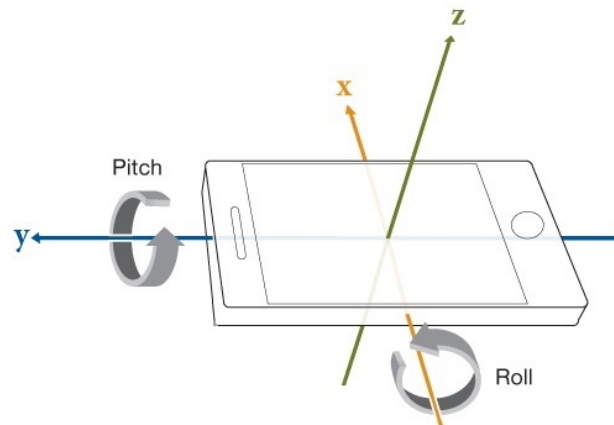


FIGURA 51. ENVÍO DE PITCH Y ROLL BTCONTROLLER

Finalmente, el envío de las señales de control se realiza, mediante el protocolo de comunicación, con la cabecera, los cuatro bytes de señal y el CRC. Véase Anexo con el código de la función de envío.

6.6. OnBoard Control

OnBoard Control es el nombre elegido para la aplicación que ejecuta el Smartphone que está a bordo del helicóptero.

A modo de síntesis, esta aplicación realiza las funciones de recibir las señales de control del mando exterior, procesarlas mediante un regulador y enviar al microcontrolador la posición de los servomotores y la velocidad de los motores.

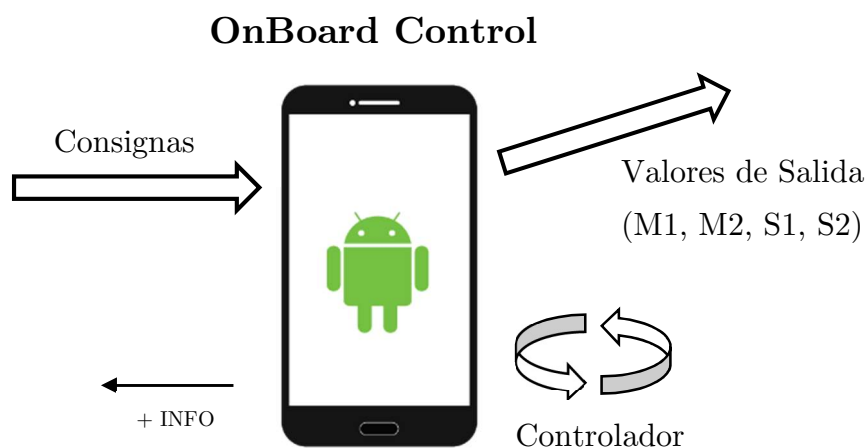


FIGURA 52. DIAGRAMA ONBOARD CONTROL

El inicio de la aplicación nos muestra una interfaz similar a la del punto anterior. Siendo la única diferencia el dispositivo a conectar, el microcontrolador.



FIGURA 53. INTERFAZ INICIO ONBOARD CONTROL

Al igual que en la aplicación anterior, en la parte inferior de la pantalla, se localiza un objeto *ListView* con todos los dispositivos vinculados por Bluetooth con el dispositivo. Se implementa así para no tener que realizar una búsqueda Bluetooth de dispositivos cada vez que se inicia la APP, mostrando solo los dispositivos vinculados con el terminal previamente.

Una vez seleccionado el microcontrolador, en nuestro caso la placa ZUM BT-328, se intenta realizar la conexión y pasar a la siguiente actividad.



FIGURA 54. INTERFAZ ACTIVIDAD DE CONTROL

La segunda actividad, mostrada en la figura anterior, es muy sencilla, sólo incluyendo dos botones.

El botón de *Logout*, arriba a la derecha, cierra los dos sockets Bluetooth, finaliza los hilos de ejecución y muestra la interfaz de inicio. Por otro lado, el botón *Start* es el encargado de empezar con todas las funciones de la aplicación.

Una vez conectado el mando exterior, al pulsar el botón, se inicia la escucha de mensajes de éste. Del mismo modo, también se ejecuta el hilo de ejecución del regulador, siendo este el responsable de analizar los datos recibidos y enviarlos al microcontrolador.

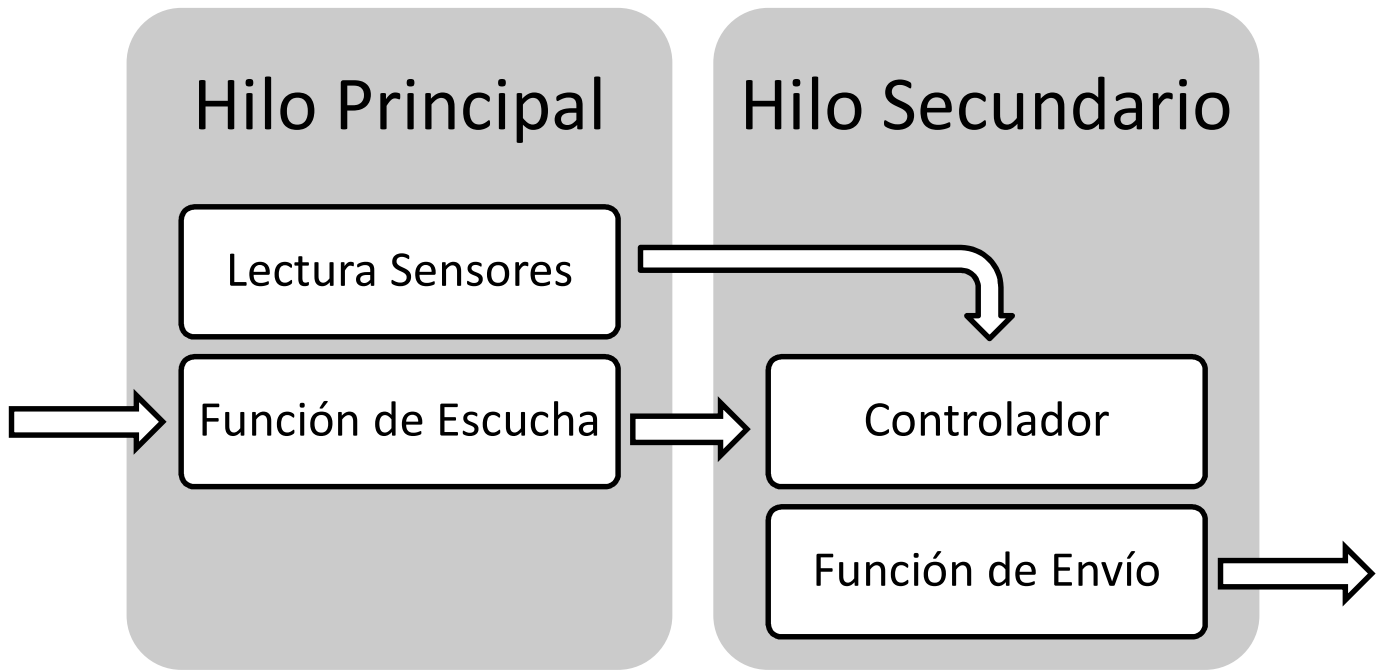


FIGURA 55. DIAGRAMA FUNCIONES ONBOARD CONTROL

Para la función de escucha de señales de control, a través del socket Bluetooth entre los Smartphones, se utiliza el protocolo de comunicación definido en apartados anteriores.

Para programar este protocolo en software, se crea un bucle que retire bytes del buffer de entrada siempre que haya disponibles. Esto es necesario ya que, si el buffer se colapsa, el teléfono móvil deja de tener la información actualizada y no responde correctamente. Cuando el byte leído corresponde con la cabecera, es señal de que los próximos cuatro bytes corresponden a un nuevo mensaje. Estos bytes no tienen por qué llegar de manera continua, por eso, se van almacenando en un vector hasta que se recibe el byte correspondiente al CRC. Una vez recibido este último byte, el siguiente paso consiste en verificar la correcta recepción del mensaje, calculando el CRC con los valores del mensaje recibido y comparándolo con el último byte del mensaje. Si el código de verificación coincide, el mensaje será correcto y se pueden dar valores a las variables de SetPoint, para su posterior tratamiento en el regulador. Véase Anexo con el código de la función de escucha.

Para tratar los datos recibidos, se implementa un hilo secundario. La misión de este *thread* es procesar las variables a través del regulador para su posterior envío al microcontrolador.

El constructor de este hilo se encarga de asignar el tipo de regulador para cada variable, pudiendo modificar los parámetros del regulador, el tiempo de muestreo y la escala de salida.

La realimentación del regulador, el valor de las variables Pitch y Roll, la podemos obtener con los sensores del Smartphone. A través del acelerómetro y magnetómetro del dispositivo, seremos capaces de obtener la orientación en el espacio del terminal y, en consecuencia, la del helicóptero.

```
public void onSensorChanged(SensorEvent event) {

    if (event.sensor.getType() == Sensor.TYPE_ACCELEROMETER)
        mGravity = event.values;
    if (event.sensor.getType() == Sensor.TYPE_MAGNETIC_FIELD)
        mGeomagnetic = event.values;

    if (mGravity != null && mGeomagnetic != null) {
        float R[] = new float[9];
        float outR[] = new float[9];

        float I[] = new float[9];
        float[] mOrientation = new float[3];

        boolean success = SensorManager.getRotationMatrix(R, I, mGravity, mGeomagnetic);
        SensorManager.remapCoordinateSystem(R, SensorManager.AXIS_X, SensorManager.AXIS_Z, outR);

        if (success) {
            SensorManager.getOrientation(outR, mOrientation);
            degrees_Pitch = (float)Math.toDegrees(mOrientation[1]);
            degrees_Roll = (float)Math.toDegrees(mOrientation[2]);
        }
    }
}
```

FIGURA 56. CÓDIGO PARA CONOCER LA ORIENTACIÓN DEL SMARTPHONE

En la anterior figura, se puede observar como con los valores del acelerómetro y magnetómetro, de los que dispone el teléfono móvil, se puede calcular su matriz de rotación para conocer la orientación en el espacio (16).

Con los datos de Pitch y Roll, calculados a través de los sensores, y los SetPoints de estos, recibidos desde el mando exterior, se ejecutan los reguladores para conseguir la estabilización del sistema. Se crea una clase PID con el código para ejecutar el controlador, pero es en la definición de los parámetros del regulador donde se eligen las acciones que incorporará, determinando el tipo de controlador. Véase Anexo con el código del hilo de control.

Por último, se tienen que enviar los datos procesados al microcontrolador para que actualice las salidas de los servomotores y motores.

Este proceso se realiza desde el hilo del controlador. Después del paso por el controlador, las variables ya están en el rango del protocolo de comunicación para el envío. Inmediatamente después se envían a la placa ZUM BT-328, actualizando el estado de los dos motores y de los dos servomotores.

Así, finalmente, el envío de las señales de salida se realiza mediante el protocolo de comunicación, con la cabecera, los cuatro bytes de salida y el CRC.

Capítulo 7

Gestión del proyecto

En este capítulo se describe la gestión del proyecto, detallando su ciclo de vida, planificación y presupuesto.

7.1. Ciclo de vida

El desarrollo del trabajo sigue una evolución similar a la representada en la redacción de este documento.

En primer lugar, al presentar los vehículos aéreos no tripulados una disciplina desconocida, se estudian sus comienzos así como la evolución hasta el estado actual. Fue necesaria la lectura de la documentación de los proyectos relacionados para obtener una visión más cercana de los objetivos a desarrollar.

Una vez estudiados los componentes necesarios para la construcción del helicóptero, se establecen las especificaciones y requisitos que se debe cumplir para una correcta elección de cada elemento.

Como se comenta en el apartado de Motivación del proyecto, es necesario introducirse en el entorno de desarrollo Arduino para ejecutar simples instrucciones y, con el tiempo, sentencias más complejas.

En paralelo, la construcción y diseño del primer banco de pruebas conlleva bastante tiempo, puesto que es imprescindible el aprendizaje de las tecnologías de impresión 3D y fresado CNC. Estas tareas, diseño y construcción, nos acompañan la mayor parte del trabajo, incluyendo mejoras cada corto periodo de tiempo.

Por último, se desarrollan las aplicaciones Android. Es necesario un estudio preliminar del entorno de desarrollo y del lenguaje, que nos facilitará abarcar todos los objetivos, como pueden ser extraer los valores de los sensores y comunicarse a través de un protocolo seguro de comunicación.

7.2. Planificación

La planificación del proyecto se muestra en el siguiente diagrama de Gantt, ilustrando la evolución temporal aproximada, descrita en el apartado anterior.

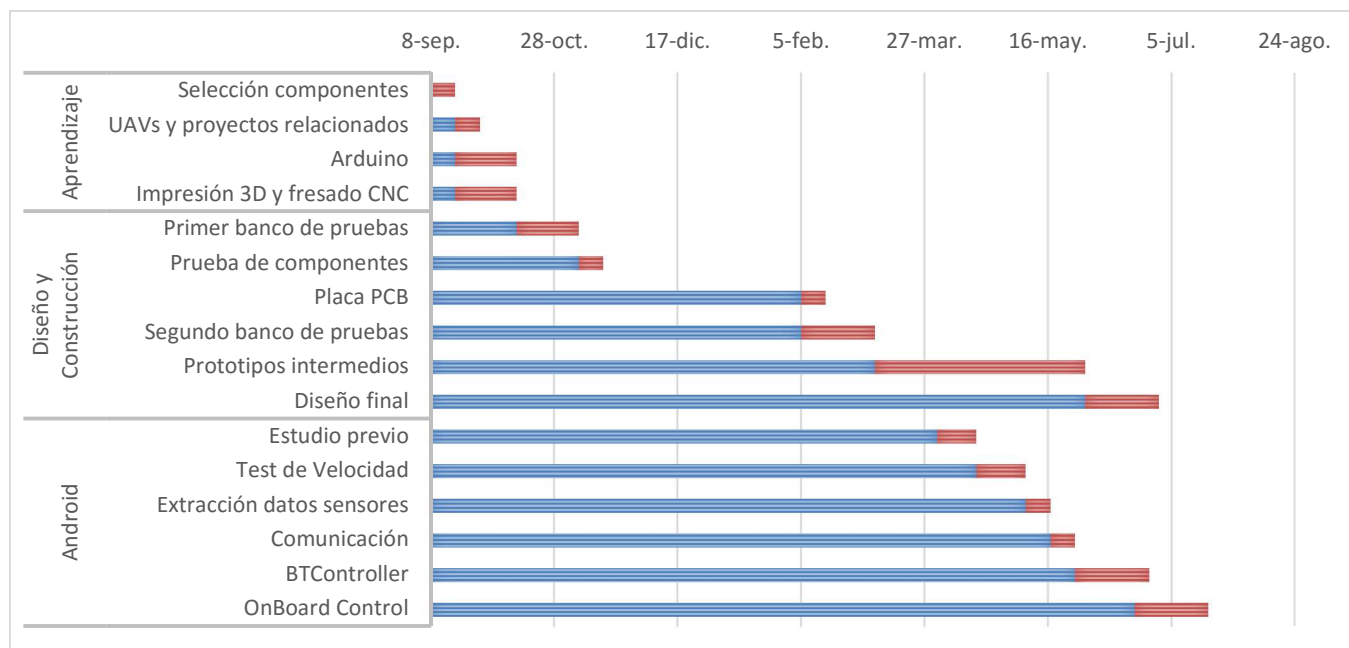


FIGURA 57. DIAGRAMA DE GANTT CON LA PLANIFICACIÓN DEL PROYECTO

7.3. Presupuesto

7.3.1. Personal

El trabajo ha sido realizado por un estudiante de Grado en Ingeniería Electrónica Industrial y Automática durante 825 horas.

7.3.2. Material

Los gastos materiales del proyecto se enumeran en la siguiente tabla.

Material	Unidades	Precio/Unidad	Precio Total
Motor Coaxial	1	53,99 €	53,99 €
Batería	1	19,06 €	19,06 €
Controladores Electrónicos	2	12,11 €	24,22 €
Servomotor	2	12,31 €	24,62 €
Microcontrolador	1	34,90 €	34,90 €
Sensor Ultrasónico	1	14,90 €	14,90 €
Filamento PLA	1	19,90 €	19,90 €
Varillas de carbono 2 mm	4	1,32 €	5,28 €
Total			196,87 €

7.3.3. Resumen de costes

Todo el software utilizado en el trabajo se ha obtenido con licencias para estudiantes y licencia Campus de la Universidad Politécnica de Madrid, por lo que no ha supuesto ningún gasto.

Los costes totales corresponden, exclusivamente, al presupuesto del material.

Capítulo 8

Conclusiones

A continuación, se presentan las conclusiones sobre los resultados obtenidos en la elaboración del presente trabajo y se exponen posibles desarrollos futuros.

8.1. Conclusión

Una vez finalizado el proyecto, se puede comprobar el cumplimiento de todos los objetivos marcados al comienzo del trabajo.

Este trabajo concluye con la construcción de un helicóptero coaxial controlado por el desplazamiento del centro de gravedad y el desarrollo de las aplicaciones Android, cumpliendo los objetivos marcados. Por un lado, se consigue la comunicación, con el protocolo de seguridad, desde el mando exterior. Y con el Smartphone a bordo, se logra la comunicación a alta frecuencia con el microcontrolador y la obtención de los datos sensoriales para obtener la inclinación de la aeronave.

Cabe destacar la exclusividad del diseño, debido a la escasez de precedentes con la característica de desplazar el centro de gravedad. Como consecuencia, el mecanismo para el desplazamiento de esta aeronave es único, al igual que la totalidad de su estructura y su protección. Por otro lado, la idea de montar un teléfono en el helicóptero como plataforma para extender sus posibles aplicaciones supone, también una idea innovadora, que no está presente en los actuales vehículos aéreos no tripulados.

Por todo ello, el desarrollo inicial de la aeronave ha resultado complejo. Han sido necesario muchos diseños intermedios para visualizar el prototipo e ir adaptando las nuevas ideas a lo ya existente. La fase de diseño, hasta conseguir

el diseño final, ha requerido una gran cantidad de tiempo que, sin duda, se hubiera podido aprovechar en el desarrollo de aplicaciones para la plataforma. En las pruebas realizadas, se ha procurado integrar las máximas medidas de seguridad, ya que una caída o rotura de algún componente lastra, y mucho, la planificación del proyecto.

En mi opinión, la base creada con el presente proyecto puede dar lugar a un sistema controlado con múltiples aplicaciones, en gran medida por la incorporación del Smartphone.

Por último, quería destacar la gran cantidad de conocimientos obtenidos con el desarrollo de este proyecto. La diversidad de competencias alcanzada cumple con creces los objetivos personales al inicio del trabajo.

8.2. Desarrollos futuros

Desde el comienzo, el presente trabajo tiene la concepción de servir de cuna para el desarrollo de gran variedad de aplicaciones.

La incorporación de un teléfono móvil en la estructura del helicóptero tiene como objetivo principal otorgar las herramientas necesarias para la adición de diferentes habilidades al helicóptero.

Cualquier teléfono actual dispone de cámara de vídeo y fotos con la que se pueden realizar grabaciones o visualizaciones en directo desde el aire. Esto puede permitir manejar el helicóptero con imágenes en primera persona o realizar fotografías y vídeos desde el aire obteniendo perspectivas diferentes.

La gran mayoría de dispositivos móviles incorporan también sistemas de posicionamiento global (GPS), que permiten obtener la posición del terminal. Estos receptores se pueden utilizar, por ejemplo, en aplicaciones para vuelo autónomo, pudiendo el operario definir trayectorias en un mapa que deba realizar el helicóptero.

Otra variante de desarrollo, diferente a la programación de aplicaciones para usuarios, es la de mejorar el control de la aeronave y aumentar su autonomía. Para ello se pueden utilizar mayor número de sensores presentes en el teléfono. Un buen control tiene que estimar la posición y orientación del helicóptero con

gran precisión. Los datos de los diferentes sensores se pueden utilizar conjuntamente para filtrar y reducir los errores, mejorando así las mediciones.

El diseño del helicóptero también puede, y debe, recibir constantes mejoras al tratarse de un producto nuevo y con un desarrollo breve. Cualquier vehículo de estas características debe ser retocado continuamente para suprimir defectos y añadir las mejoras que se observen en las diferentes pruebas que se realicen con los prototipos.

Bibliografía

1. **Ortega Biber, Eduardo.** *Diseño y construcción de un helicóptero coaxial.* Universidad Politécnica de Madrid, Escuela Técnica Superior de Ingeniería y Diseño Industrial. 2015.
2. **Wikipedia.** Vehículo aéreo no tripulado. 2015.
https://es.wikipedia.org/wiki/Vehículo_aéreo_no_tripulado.
3. **Barcala Montejano, Miguel Ángel.** *Helicópteros Teoría y Descriptiva.* Universidad Politécnica de Madrid, Escuela Universitaria de Ingeniería Técnica Aeronáutica. 2008.
4. **Coleman, Colin.** *A survey of theoretical and experimental coaxial rotor aerodynamic research.* s.l. : NASA TP-3675 Technical Paper, 1997.
5. **Petrosyan, Eduard.** *Aerodynamic features of coaxial configuration helicopter.* 2007.
6. **Ruddell, Andrew J.** *Advancing blade concept development test program.* 1981.
7. **Chen, Li.** *Modelling the lama coaxial helicopter.* 2007.
8. **Samir Bouabdallah and Roland Siegwart.** *Design and control of an indoor coaxial helicopter.* Technical report, Autonomous Systems Lab ETH Zurich. 2006.
9. **Christian Bermes, Kevin Sartori, Dario Schafroth, Samir Bouabdallah and Roland Siegwart.** *Control of a coaxial helicopter with center of gravity steering.* Technical report, Autonomous Systems Lab (ASL), Swiss Federal Institute of Technology Zurich. 2008.
10. **Zare Seisan, Farid.** *Modeling and Control of a Co-Axial Helicopter.* PhD thesis, University of Toronto. 2012.
11. **James Paulos and Mark Yim.** *Flight performance of a swashplateless micro air vehicle.* Technical report, University of Pennsylvania. 2015.
12. **Flyability.** Introducing gimball, the collision-tolerant drone.
<http://www.flyability.com/product/>.

13. **Aera. Flone.** <http://flone.aeracoop.net>.
14. **Arduino.** <https://www.arduino.cc>.
15. **Módulo Bluetooth BQ ZUM BT-328, Comandos AT.**
http://diwo.bq.com/wp-content/uploads/2014/11/BLK-MD-BC04-B_AT-COMMANDS.pdf.
16. **Android Developers.** <http://developer.android.com/develop/index.html>.

Apéndice A

Anexos

Aplicación BTController

➔ CÓDIGO DE LA FUNCIÓN DE ENVÍO

```
protected void SendPDU(int BYTE1, int BYTE2, int BYTE3, int BYTE4) {

    BufferMsg[0] = (byte) HEADER;           //Header

    BufferMsg[1] = (byte) BYTE1;           //Altura
    BufferMsg[2] = (byte) BYTE2;           //Yaw
    BufferMsg[3] = (byte) BYTE3;           //Roll
    BufferMsg[4] = (byte) BYTE4;           //Pitch

    BufferMsg[5] = (byte) ((BufferMsg[1] + BufferMsg[2] + BufferMsg[3] + BufferMsg[4]) %
                           BYTE_SIZE);    //CRC

    if (!Arrays.equals(BufferMsg, Last_BufferMsg)) {
        if (BYTE1 >= BYTE_MIN && BYTE1 <= BYTE_MAX && BYTE2 >= BYTE_MIN && BYTE2 <= BYTE_MAX
            && BYTE3 >= BYTE_MIN && BYTE3 <= BYTE_MAX && BYTE4 >= BYTE_MIN && BYTE4 <= BYTE_MAX)
        {

            sendData(BufferMsg);

        } else msg("Fallo enviando el paquete: Rango erroneo.");
    }

    System.arraycopy(BufferMsg, 0, Last_BufferMsg, 0, BufferMsg.length);

}
```

Aplicación OnBoard Control

➔ CÓDIGO DE LA FUNCIÓN DE ESCUCHA

```
private void ListenPDU()
{
    if (SocketAndroid != null) {
        final Handler handler = new Handler();
        stopListenThread = false;

        try {
            mInputStream = SocketAndroid.getInputStream();
        } catch (IOException e) {
            e.printStackTrace();
        }

        listenThread = new Thread(new Runnable()
        {
            public void run()
            {
                while(!Thread.currentThread().isInterrupted() && !stopListenThread)
                {
                    try
                    {
                        while(mInputStream.available() > 0) {
                            incomingByte = mInputStream.read();

                            if (incomingByte == HEADER) {
                                reading = true;
                                bytesRead = 0;
                            }
                            else if (reading) {
                                // byte read after header
                                if (reading && bytesRead < (N_BYTES)) {
                                    readBufferPDU[bytesRead] = incomingByte;
                                    bytesRead++;
                                }
                                // reading CRC + interpreting data
                                else if (reading && bytesRead == (N_BYTES)) {

                                    readBufferPDU[bytesRead] = incomingByte;
                                    int err = interpretData();

                                    if (err == 0) {
                                        VelSP = readBufferPDU[0];
                                        YawSP = readBufferPDU[1];
                                        RollSP = readBufferPDU[2];
                                        PitchSP = readBufferPDU[3];
                                    }
                                    else if (err == -1) {
                                        handler.post(new Runnable() {
                                            public void run() {
                                                salida.setText("ERROR RECIBIDO: CRC no coincide. " + incomingByte);
                                            }
                                        });
                                    }
                                    bytesRead = 0;
                                }
                            }
                        }
                    }
                }
            }
        });
    }
}
```

```

        reading = false;
    }
}
else {

    bytesRead = 0;
    reading = false;

    handler.post(new Runnable() {
        public void run() {
            salida.setText("Byte recibido sin header." +
                           incomingByte);
        }
    });
}

}

}
catch (IOException ex)
{
    stopListenThread = true;
}
}

});
listenThread.start();

} else {
    Toast.makeText(getApplicationContext(),
        "Bluetooth is not connected", Toast.LENGTH_SHORT).show();
}
}

//-----//
//                               Interpretation of PDU                               //
//-----//

public int interpretData() {
    int crc;

    crc = ((readBufferPDU[0] + readBufferPDU[1] + readBufferPDU[2] + readBufferPDU[3]) %
        BYTE_SIZE);

    if (crc == readBufferPDU[N_BYTES]) {

        //Paquete correcto
        return 0;
    }
    else {
        //Paquete incorrecto
        return -1;
    }
}
}

```


➔ CÓDIGO DEL HILO DE CONTROL

```
private class ExecutePID extends Thread {
    private volatile boolean close = false;
    int SampleTimeGlobal = 1; //in milliseconds

    final Handler handler = new Handler();

    public ExecutePID() {

        Pitch.setTunings((float) 1.0, 0, 0);
        Pitch.setSampleTime(SampleTimeGlobal);
        Pitch.setOutputs(-30, 30);
        Pitch.setMappings(0, 250);

        Roll.setTunings((float) 1.0, 0, 0);
        Roll.setSampleTime(SampleTimeGlobal);
        Roll.setOutputs(-30, 30);
        Roll.setMappings(0, 250);
    }

    public void run() {

        while (!close) {

            long now = System.currentTimeMillis();
            long timeChange = (now - lastTime);

            if (timeChange >= SampleTimeGlobal) {

                handler.post(new Runnable() {
                    public void run() {
                        VelM1Byte = VelSP;
                        VelM2Byte = VelSP;
                        RollByte = Roll.Controller(RollSP, degrees_Roll);
                        PitchByte = Pitch.Controller(PitchSP, degrees_Pitch);

                        SendPDU(VelM1Byte, VelM2Byte, RollByte, PitchByte);
                    }
                });

                lastTime = now;
            }
        }

    }

    public void close() {
        this.interrupt();
        close = true;
    }
}
```